Adversary Modelling: Evaluating the feasibility of symbolic adversary model on smart transport ticketing system

Sheung Chi Chan

Technical Report

RHUL–ISG–2015–2 (RHUL–MA–2015–2)

4 March 2015

Information Security Group
Royal Holloway University of London
Egham, Surrey, TW20 0EX
United Kingdom

Student Number: 100764006
Student Name: Sheung Chi Chan

Adversary Modelling: Evaluating the feasibility of symbolic adversary model on smart transport ticketing system

Supervisor: Professor Keith Mayes

Submitted as part of the requirements for the award of the

MSc in Information Security

at Royal Holloway, University of London

I declare that this assignment is all my own work and that I have acknowledged all quotations from published or unpublished work of other people. I also declare that I have read the statements on plagiarism in Section 1 of the Regulations Governing Examination and Assessment Offences, and in accordance with these regulations I submit this project report as my own work.

Signature: Date:

Abstract

Nowadays, smart card has already been adopted in many public transport ticketing system. This make the security and mutual authentication of the card and reader becomes an important factor to consider. In order to prove the security level and the security properties needed in the communication, formal symbolic adversaries modelling approach are analysed and illustrated in this project and a new approach is proposed. The proposed model base on the Process Algebra CSP model with additional consideration on the cryptographic algorithm and the change of the adversary knowledge set. This can give a more thorough analysis and consideration on the whole network environment. Three protocols will be used as the target of illustration for the proposed modelling approach. They include the original MiFare Classic authentication protocol which some other researchers dismantle it in some of the papers. Also, there exists some protocol proposed by other researchers to improve base on the original authentication protocol after the discovery of flaws in the original one. Two of them are chosen in this paper for the illustration. There will be a brief description of those protocols before the illustration of the new approach for modelling. The primary target for the new modelling approach is to consider a complete running network environment of the smart transport ticketing system and hope to have a formal way to study this kind of contactless communication of smart ticketing system protocol implementation.

Contents

1 Introduction **1**
- 1.1 Introduction . . . 1
- 1.2 Motivation . . . 1
- 1.3 Objective . . . 2
- 1.4 Structure . . . 2

2 Smart Card and Smart Ticketing **4**
- 2.1 RFID Proximity Smart Card . . . 4
- 2.2 MiFare® . . . 5
  - 2.2.1 Structure and Standard . . . 5
  - 2.2.2 Memory and Access Control . . . 7
  - 2.2.3 Functions and Security Features . . . 7
  - 2.2.4 Communication Process . . . 8
- 2.3 FeliCa® . . . 10
  - 2.3.1 Structure and Standard . . . 10
  - 2.3.2 File System and Controls . . . 11
  - 2.3.3 Functions and Security Features . . . 13
- 2.4 Cipurse and Calypso . . . 14

3 Adversaries of Smart Transport Ticketing System **15**
- 3.1 Authentication Protocol . . . 15
  - 3.1.1 Nonce and Response . . . 15
  - 3.1.2 Crypto1 Encryption . . . 16
- 3.2 Security Properties, Adversaries and Attacks . . . 17
  - 3.2.1 Confidentiality . . . 17
  - 3.2.2 Integrity . . . 18
  - 3.2.3 Availability . . . 19
  - 3.2.4 Weakness on Authentication Protocol . . . 19
  - 3.2.5 Weakness on Implementation . . . 20
- 3.3 Improved Protocol . . . 21

4 Adversary Modelling **23**
- 4.1 Symbolic Adversary Modelling . . . 23
- 4.2 Process Algebra CSP Model . . . 23
  - 4.2.1 Process Algebra Language . . . 24
  - 4.2.2 Signalling . . . 24
  - 4.2.3 FDR and Casper . . . 25
  - 4.2.4 Rank Function . . . 26
- 4.3 Dolev-Yao Model . . . 26

5 Adversary Modelling on Smart Transport Ticketing System **28**
- 5.1 CSP Modelling Language . . . 28
  - 5.1.1 Processes and Events . . . 28
  - 5.1.2 Actors, Systems and Network Environment . . . 30
  - 5.1.3 Trace Logic . . . 31
- 5.2 Adversaries Knowledge Consideration . . . 32
  - 5.2.1 Knowledge Deduction and Generation . . . 32
  - 5.2.2 Affect to Rank Function Theorem Proving . . . 33
- 5.3 Proposed Solution . . . 34

6 Analysis the Proposed Adversary Model **35**
6.1 Illustration of Proposed Model 35
6.1.1 Original MiFare Classic Authentication Protocol 35
6.1.2 Improved Protocol in [HC12] 38
6.1.3 Improved Protocol in [JK13] 41
6.2 Analysis of the Proposed Model 43

7 Conclusion **45**

1 Introduction

1.1 Introduction

Smart Card, a pocket-sized plastic card with memory, microprocessor and an embedded circuit, aim to provide fast, portable, reliable, secure and convenient application process and data storage. The advantages of smart card have attracted different business sectors to adopt it as new media supporting their traditional services. For example, the chipped bank card using in the banking industry and the subscriber identification module, or SIM card in short, used in the mobile telecommunication services.

In the development of public transportation system, ticketing is always an important factor to be consider. Normally, travellers need to purchase tickets before using the public transportation service in order to prove that they had paid for the journey that cover the operation cost and revenue of the operating company. Occasionally, the traveller is unwilling to pay for the service and may try to avoid or reduce their payment in order to gain benefit for their own. This action directly affected the income of the operating company and may result in a rise of the ticket price, which is unfair to typical travellers that have to withstand the loss from those selfish people who skip or reduce the fare by a fraud ticket or else. In order to minimize the incidents of the fraud ticket, many transport operators start to adopt electronic ticket to solve this problem. One of the general electronic ticket is the magnetic stripe ticket which is readable by machine. Although it helps to reduce the number of the fraud ticket and the enormous need of human inspection of paper tickets, there are still lots of extra items and features wanted by the transport operators. They want the ability to track the transport record with a fast, secure and reliable transaction and the ability to disable a ticket in the case of lost or fraud of the ticket. These requirements lead to the adoption of smart card on the transport ticketing system in recent year.

It is no doubt to say that the implementation of smart card in transport ticketing system has many advantages, but the security concern and technical vulnerability of the smart card protocol used are a significant problem to consider. The preliminary reason for adopting electronic ticketing in public transportation system is to reduce fraud and unfair price reduction in the public transport. So if someone breaks the security constraint of smart card based ticket and successfully created a fake card to use as a genuine electronic ticket, then the reason for using smart card on public transport ticketing become insignificant. For this reason, the security level of the smart card used in the public transportation ticketing needed to be maintained and seriously protected.

1.2 Motivation

In the world of information security, modelling is a technique and framework to help to make the protocol and system become easier for understanding and analysis. With a better understanding, it is easier to identify potential problems and vulnerabilities in the protocol and system and fix it in advance of incidents. This action can undoubtedly increase the security level of the system. There are many modelling techniques, one of them is the adversary modelling, which identify the behaviour of the adversaries (attackers). Also, it can help to identify the competitiveness, reliability and security level of a system with the present of adversaries who try to eavesdrop and intercept the target system and protocol with their knowledge and computational power.

In the case of smart transport ticketing system, the security level of the protocol and system use is the main concern, and currently, the usage time of smart card in transport ticketing system is still very short. Although there are already lots of adversary modelling frameworks exist in the field, there are very few of them target to the limited computational power of the smart card protocol and system that used on the public transport ticking. In order to have a more scientific way to understand and study the security of smart transport ticketing system, there are needs to review the existing adversary model. The study can help to see if it is feasible to model and help to analysis the protocol and system by formal modelling in order to raise the security level of the smart transport ticketing system.

1.3 Objective

This project aims to review on some common smart transport ticketing systems and the attack target to them. Also, some general adversary models will also be reviewed to prepare for the feasibility study of their usage on the modelling of smart transport ticketing system. The main objective can be classified as follows:

1. Find and propose a symbolic adversary modelling framework that are suitable for the analysis of the smart transport ticketing system and protocol
2. Investigate the ways to extend the model to allow it become possible to analysis some non-logical part of the smart transport ticketing system and protocol
3. Use formalized model to analyse smart transport ticketing system to help to identify the potential problems, weaknesses and vulnerabilities of the system and protocol, which can help to provide a solution to mitigate attacks.
4. Analysis the feasibility of the proposed model on smart transport ticketing system.

1.4 Structure

This document is structured as follows:

- Chapter 2, titled "Smart Card and Smart Ticketing", provides some background information of smart card and the RFID technology which is used in the smart transport ticketing system. Also, it provides some overview and description of some existing RFID proximity card chips used in smart transport ticketing system.
- Chapter 3, titled "Adversaries of Smart Transport Ticketing System", provides some technical details and description on some existing weaknesses, adversaries and attacks targeted to some of the smart transport ticketing system. Also, some fixed protocol will be discussed.
- Chapter 4, titled "Adversary Modelling", provides some background information of formal modelling technique and some existing symbolic adversary modelling framework.

- Chapter 5, titled "Adversary Modelling on Smart Transport Ticketing System", provides a proposed adversary modelling approach for using on the analysis of smart transport ticketing system. Also, some description and how it extends from the existing adversary modelling framework are illustrated.
- Chapter 6, titled "Analysis the Proposed Adversary Model", provides an example illustrating how the proposed adversary modelling approach work on analysing the smart transport ticketing system. Also, a brief analysis of the proposed approach are illustrated.
- Chapter 7, titled "Conclusion", provides a brief summary of the whole adversary modelling analysis on smart transport ticketing system.

2 Smart Card and Smart Ticketing

In a few decades ago, the banking industry has started to adopt magnetic strip card into their business in order to replace the need of the human teller by some automatic teller machines. The magnetic strip loaded with customer data, together with a plastic card body form an early electronic card. The preliminary purpose of these kinds of electronic memory card is to identify the person presenting the card to the machine (and back-end system) which are trying to read the data from the card.

Smart card is a kind of electronic card intended to provide different usage, storage and computational power when comparing to magnetic strip card. It stores the data in the silicon chip which embedded in the card. In the context of smart card, there are two main categories. The first category is memory card that store information and provide preliminary control on access of data blocks. The other category is the processor card, which have one or more microprocessor embedded inside the chip to perform different instructions and operations on top of the data stored inside the memory of the chip. In this case, smart cards can be treated as a microcomputer that can perform limited operations on the data stored inside the card, including access control and data protection through encryption and signatures. In additional, the operations can be used for simple authentication in order to provide an extra layer of security or act as user-owned media of multi factors authentication. Microprocessor or wired-logic, which located inside or around the chip with computational power, will provide all these functions of smart card.

2.1 RFID Proximity Smart Card

For contactless smart card, the silicon chip embedded inside the card will need to power up and contact through electromagnetic wave like the radio wave. Some of them make use of the fast and reliable communication of radio frequency and implement RFID system on contactless smart card. This action aimed to extend the use of RFID system on the smart card. In the other word, it tries to preserve the contactless radio wave communication and identification and provides additional operations and security control. These kinds of contactless smart card divide into two groups. The first group is passive card, which named as RFID proximity smart card and described under ISO14443 standard [ISO11]. This kind of card does not have the power on itself and totally depends on the readers to provide service power by radio wave and electromagnetic induction on antenna. The second group is the active card, which named as RFID vicinity smart card and described under ISO15693 standard [ISO09b]. It has batteries installed inside the card that allow the card to have its own computationi and operation power. Because of this reason, this kind of card can work in a longer distance because it does not require near field electromagnetic induction for powering up its chips and processors. In this project, only RFID proximity smart card will be reviewed in deep as it is the primary type of smart card used in transport ticketing system.

RFID proximity smart card makes use of the radio wave communication of the RFID system as the communication interface between the chip and the readers. The integrated circuit, antenna and chip (may contain one or more microprocessor) inside the card will act as the transponder. When the card enters the interrogation area of the card reader, they can start to communicate and provide the basic identification features of RFID and other additional access control, authentication, and data protection such as the three ways handshaking mutual authentication. This kind of RFID proximity smart card can usually store many data, including electronic purse balance or other sensitive information and can

provide fast and reliable transaction. Because of that, it is commonly used as electronic purse (or wallet) and transport ticketing which cannot withstand significant delays in some of the cases. The ISO14443 standard covers information of RFID proximity smart card, which contains details technical descriptions of the card. The standard provides two types of card (ISO14443-A / ISO14443-B) which have different communication protocol. This project aimed to analysis smart transport ticketing system and RFID proximity smart card based on ISO14443-A, which is the primary type of smart card used in the transport ticketing system.

In smart transport ticketing system, contactless wired-logic smart card, which is a member of RFID proximity smart card, is mainly used. In the following section, some major chips system brand of RFID proximity smart card using in smart transport ticketing system is briefly reviewed.

2.2 MiFare®

NXP, formally part of Phillips, is the inventor of the MiFare system. MiFare system family has four notable brand groups, namely MiFare Classic, MiFare Plus, MiFare Ultralight and MiFare DESFire. They combine to occupy a large market share in the Proximity smart card field. The target applications of the MiFare family is on public transportation ticketing, access control management, electronic purse and payment, loyalty card and many other services.

In global transport ticketing system, many of the transport operator has adopted the MiFare family chips in their electronic ticketing implementation. For example, the Oyster Card[1] in London, United Kindom, the OV-chipkaart card[2] in the Netherlands, the Easy-Card[3] in Taiwan, and the Yikatong Card[4] in Beijing, China. In the MiFare family, the most commonly used variant is the MiFare Classic, which is the first product line of the MiFare family. Although all of its security primitive is secret when release, there are still some researchers successfully reverse engineered the design of MiFare Classic and found loads of weaknesses in it and discovered several possible attacks on it. Since then, many of the transport operator has switched the usage of MiFare Classic chip to a lately developed and more secured MiFare Plus or MiFare DESFire. Indeed, MiFare Classic is still the major chips brand used in the MiFare family worldwide. This project aimed to analysis the modelling of existing smart transport ticketing system, so MiFare Classic has been chosen for the study because of its wide usage and the existing of many weaknesses and possible attacks. The following subsection aimed to review the basic structure and design of MiFare Classic. Also, some review of its weaknesses and practical attacks exists in order to help to analysis the feasibility of symbolic adversary modelling on a widely used smart transport ticketing system (MiFare Classic) will be presented in the next chapter.

2.2.1 Structure and Standard

MiFare Classic follows the standard of ISO14443 Type A. It provides chips will two different size of EEPROM (1KByte and 4KByte) [NXP11a, NXP11b]. The two major components in MiFare Classic system include the Proximity Coupling Device, or PCD in short, which

[1] https://oyster.tfl.gov.uk/
[2] https://www.ov-chipkaart.nl/
[3] http://www.easycard.com.tw/
[4] http://www.bmac.com.cn/

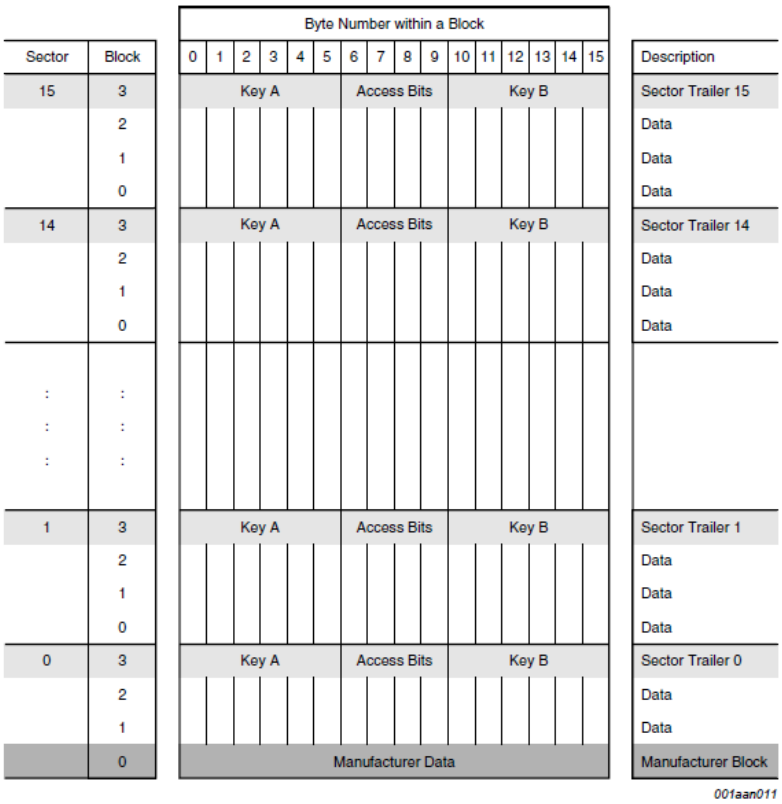

Figure 1: Memory structure of MiFare Classic 1K chip

is usually present as a form of RFID reader and writer that connected to a terminal or PC to provide service to the MiFare Classic card. The other component is the Proximity Integrated Circuit Card, or PICC in short, which includes the antenna and MiFare Classic chip on the smart card side. It will passively power up when entering the interrogated area of the PCD (usually in a range of 10cm) and start communicating with the PCD to control the data stored in the chip to provide services. This radio frequency communication interface follows all the requirement mentioned in ISO14443-A standard.

The MiFare Classic can be classified as a memory chip with extra security features to control those data stored in the chip. In basic, the most important part of the MiFare Classic chip is the EEPROM memory which store all the data values and all the security control primitives and keys. The main different of the two variants 1K and 4K of MiFare Classic is on the size of the EEPROM. All other security measures and features are exactly the same. For the logical structure of the EEPROM, 1K chip divided equally into 16 sectors of four data blocks with 16 bytes in each block. In contrast, 4K chip will divide the first half of 2KByte equally into 32 sectors of four data blocks with 16 bytes each, and then divided the second half of the memory equally into eight sectors of 16 data blocks with 16 bytes each. For both the 1K and 4K EEPROM, the first block of the first sector is reserved as the Manufacturer Block and the last block of each sector is reserved as the sector trailer which aimed to store the keys and access control data for that sector. All remaining blocks are used to store data values. Each chip will have a seven bytes unique identifier stored in the head of the Manufacturer Block for unique identification. It also aimed to provide card specific security control and anti-collision control. Figure 1 shows the structure of the memory for 1K MiFare Classic chip.

2.2.2 Memory and Access Control

MiFare Classic has two different memory size (1KByte and 4KByte) with a little bit different in the blocks and sectors division. Except the division, all other features like the memory access command and control remain the same for the two variants. It provides six different memory access operations, includes Read / Write operations that allowed to process on all block except block zero of sector zero which is the manufacturer block that are only allow to write once and read many times. The remaining four operations are increment, decrement, transfer and restore which only allowed on value block and will not affect the content on sector trailers, which contains the keys and access control bits. All the memory operations are further controlled by the keys and access control bits stored in the trailer of each sector, providing control for its sector. As a result, each sector is allowed to have its secret keys and access control bits to control the memory access and services.

In MiFare Classic, 48-bits sector keys are used. There will be at most two keys for each sector located at the sector trailer to control that sector. In the sector trailer, the first six bytes are used to store key A, which is never allowed to read in any case. The last six bytes are used to store another key B, which is optional in some of the case that no memory operation needs to use key B to gain access to the memory block. In the case of key B not exist, the six bytes can be used to store regular data when the common value block is full. The four bytes in the middle are used to store the access bits. There will be three access bits for each of the blocks, which aimed to control the access of the three data blocks and the section trailers. So there will be four sets of access bits store in the sectors trailers which occupy the three bytes in the access bits area of the sectors trailer. The remaining byte are reserved for other data (In the case of the last eight sectors of the 4KByte MiFare Classic variation, the four sets of access bit group will not only control one block of data per set. Indeed, it will control group of data blocks that this block will share the same access control elements).

As each group of access bits contains three bits, there will be eight combinations of the access condition for each sector. Each combination defines which of the six memory operations are allowed in this block. If the operations are allowed, which keys (key A and key B of the sector that this block belongs to) are required for encryption and decryption of the commands, operations and data for the memory operation which aimed to access or modify the memory data will also be stated. For the access bits controlling the sector trailer, it also has the same meaning; the eight access bits combination are used to control the read-write of the two keys and the access bits, indicating which key and access bit are allowed to read or write. Also, it indicates that which key will need for the key and access bits read-write actions. The operations on the sector trailers always aimed to replace keys and access control of the memory blocks.

2.2.3 Functions and Security Features

Being a kind of RFID proximity card, MiFare Classic provides some functions and security features on top of the simple identification purpose of typical RFID system and regular memory card. From the previous subsections, it mentioned that MiFare Classic will use the keys and access bits in the sector trailers to control the memory operations and access. The memory operations are encrypted and decrypted by the key, which is one of the parameters send to the cryptosystem named as CRYPTO1. Although NXP keeps the design details of CRYPTO1 as a secret, there are already a lot of researchers successfully

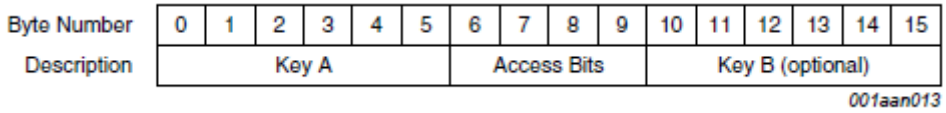

Figure 2: Structure of MiFare Classic sector trailers

reversed the CRYPTO1 algorithm and found some weaknesses in it [Tan09, SVRG+08]. Some researchers even success to launch some practical attack by exploiting the weaknesses of CRYPTO1.

Apart from the data security consideration, MiFare Classic also provided with an anti-collision feature which aimed to select only one card for communication. This setting is required when there are more than one card in the interrogating area of the same card reader which are activated by the broadcast request sent from the reader. The select card procedure are defined on the ISO14443-A-3 standard. After the selection of card, the reader will start the mutual authentication process with the card with the selected ID (which should be unique). Other cards inside the interrogation zone of the reader will back to the idle state and wait for the next request command broadcast by the reader. The card will become active again for anticollision loop and communication process.

In additional to data security and anti-collision feature, MiFare Classic also provides some measures and mechanisms to check for data integrity in order to maintain a reliable channel for radio frequency communication between the reader and the card which contains high amount of noise and interference. In each of the data stored in the memory of the card and data sent between reader and the card, there will be a 16 bits CRC attached to each block of data. Also, parity bits will also be included in each byte of data. In addition, some bits counting will be done on data to ensure the data sent and received has the same number of bits. As most of the data communication through the radio frequency is in broadcast type, so it is unexpected that when will data sent or received in the communication channel. The case of sending a serial of zero value (which has no power sent in some of the encoding methods) and the case of sending nothing needed to be separated in order to distinguish the case of sending zero as the data or no information is sent. This separation is also a feature provided by MiFare Classic. In most of the data storage of EEPROM in MiFare Classic, the same block of data is stored more than one time in the memory. For example, in any value block of 16 bytes, the signed value of four bytes will be stored three times, the standard value will save on byte 0-3 and byte 8-11, and the inverse of the value will store in byte 4-7. Even the access bit in the sector trailers will have redundancy that keep the standard form and inverse form of the access bit in the memory together. This redundancy can help to recover data in the case of partial memory corruption and ensure the data integrity and reliability in the memory. Figure 2 shows the structure of the sector trailer.

2.2.4 Communication Process

MiFare Classic follows the communication defined in the part 3 of the ISO14443-A standard. Also, some additional steps have been self-described in MiFare Classic to provide a more secure and reliable communication on top of the standard. As the chip of MiFare Classic is passive, it needs the reader to give it power to run. When the card is out of the range of the interrogating field of the reader, it will remain in Power On Reset (or POR in short) state. Then it will wait for the carrier of the card to re-enter the interrogating area of the reader and receive a request command broadcast from the reader frequently. The

reader will transmit the request command (or request all command to "wake up" cards in the HALT state), then wait for a predetermined time and repeat until receiving Answer To Request which sent from the chip of the MiFare Classic card. The chip will sent that Answer To Request when it has enter the interrogation area of the reader and receive the request command broadcast. Once the reader has received one or more response from the card, it will stop the broadcast of request and start the anti-collision loop in order to determine the card granted for this communication round to ensure a one to one communication relationship could be established.

When the communication process is in the anti-collision loop stage, the reader will try to resolve the collision and ultimately allow only one card to establish a communication channel with the reader. The collision is detected by examining the encoding signal (Modified Miller or Manchester Coding, which defined in the ISO14443-A part 3 standard) sent from the cards, contains their UID. If there is more than one card sending the signal response, than there must exist at least 1 bit of complement value, and the resulting signal will be affected and become unreadable in the bit of crash. In this case, the reader will mark down the location of the collision bits, and the remaining bits is read as normal. After that, the reader will send a new request to limit the size of respondents by defining an accepted UID range, which aimed to define exact allowed value for the first collision bit. Then the range of allowed UID will be decreased. All the active card tags within the allowed range will send another response to the reader, all the other active card tags will return to the standby state which will become active again when a new request broadcast sent from the reader is received. The loop will continue until no more collision is found. That is only one card is responding to the reader request. In this situation, the reader will send a select card command (with the UID of the selected card) to notify the card that it has been selected for this communication and is allowed to establish a connection with the reader and start the authentication stage.

After the communication channel of the card and the reader established, the memory access operations and commands will be allowed. In MiFare Classic, each sector of the memory is secured and controlled separately by the keys and access bits stored in the sector trailer. So whenever the reader wants to access and operates on a sector, authentication required. The authentication will also be needed when the reader finishes operations on one of the sectors and needed to operate on other sectors. MiFare Classic uses three-way mutual authentication to prove that both parties in the communication have knowledge of the key without passing the key through the communication channel. The details of the authentication protocol will be illustrated in the next chapter.

After the authentication, the reader can operate, access or control the memory according to the settings of the access bits. When the reader needs access to different sectors, the re-authentication process using different keys will be needed. In any time in the communication process, the reader can send a HALT command to the reader, which will aimed to stop the communication between the reader and the card. In this situation, the card will move to the HALT state, which will not response to any request broadcast from the reader. If the card keeps staying in the interrogation area of the reader, the power of the card will not lost. In this case, only a Wake Up command (which will active all the card in the interrogation area of the reader to response to the reader's request) will wake the card tag up. If the card leaves the interrogating area of the reader, it would loss power and will back to the POR state and will response to the reader when it re-enter the interrogating area of the reader.

The project is aimed to analysis the feasibility of symbolic adversary modelling on the protocol of smart transport ticketing system. So the project will not review too deep into the detailed design which can be found in the official specification [NXP11a, NXP11b] and some of the research papers [MC10, Cri09].

2.3 FeliCa®

The inventor of FeliCa chip is Sony. They provide three families of contactless smart card; they are the FeliCa IC Card, FeliCa FRAM Card and the FeliCa EEPROM Card. FeliCa IC Card [Son12] is the variants with the most user. It is less distributed if compare to the MiFare system in the usage of contactless smart card. The target applications of the FeliCa family is on public transportation ticketing, electronic control token, electronic purse and payment, loyalty card and many other services. In the preliminary design of FeliCa Card, it aimed to support multiple applications, which can use one card for many services. For example, a FeliCa card with electronic purse function can also add transport ticketing function in it, which can cooperate with the electronic purse and deduct money from it on each ticketing. Also, it can be used as a regular payment method, and even add loyalty point scheme in it and earn loyalty point in each payment. At last, it can also add features for access control or act as a token for some electronic devices or appliances. Its design aimed to run multiple application in cooperate mode in one physical card.

Two of the largest user group of FeliCa card is the Octopus Card[5] in Hong Kong and the cooperative card[6] in Japan. The Octopus Card is an excellent example of real implementation of multiple applications run in cooperation mode. In Hong Kong, most of the people have at least one Octopus Card in their hand. Yhe main purpose is to serve as an electronic wallet with the prepaid money stored inside the card. Many locations like restaurants, entertainment venues and supermarkets support Octopus Card payment (direct deduction of the card's remaining value). Another significant usage is the transport ticketing system, which the fare for the transportation deduct from the card directly. The card can act as a prove of payment without a need for an extra ticket during the journey. Nearly all form of transportation in Hong Kong support payment by Octopus Card. Also, companies, residential management or other organization willing to share the use of Octopus Card can paid them and request for a reader to be installed as access control and other identification purposes. They just need to add some identification token (or some apps) in the target user's card and the Octopus Card can support those features without a replacement of the card.

2.3.1 Structure and Standard

FeliCa IC card also follows the standard of ISO14443 Type A, which is as same as the MiFare family. For the three variants, they provide different kind of storage, some with FRAM and some with EEPROM. Like MiFare family, there are two major components in FeliCa system includes the transponder, which is the card chips and an antenna, and the control board (reader) that aimed to power up the chips and transmit information. There are two encoding type mentioned in the RFID proximity contactless smart card standard (ISO1443-A, which is the standard followed by both MiFare and FeliCa family), they are

[5] http://www.octopus.com.hk/

[6] This is a project to let the 10 major FeliCa base IC card system of separate city to become interoperable and can be used throughout Japan

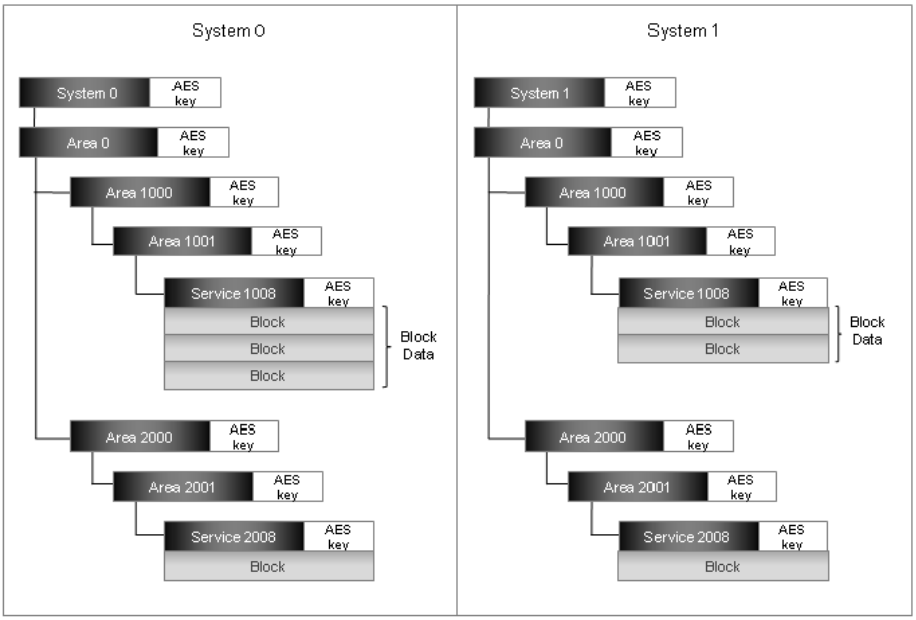

Figure 3: Example of FeliCa memory with AES key

the Modified Miller or Manchester Coding. In FeliCa, Manchester encoding scheme is chosen which aimed to tolerant a higher amount of noise that the communication channel between the card transponder and the control board. Due to the portable design of the antenna and chips, the actual shape of the card is very flexible. The implementation of Octopus Card has many different shapes, and some Octopus Card can even embed in mobile phones and watches.

2.3.2 File System and Controls

In FeliCa IC card, it manages the storage by file system; it is different when comparing to the simple blocks and sectors division in the MiFare Classic memory. In FeliCa file system, it provides three levels of hierarchy component to control the data block. The top of the hierarchy is the System, the next level is the area. In the area, it will contains multiple services, and each service will maintains it own data block of storage. The file system stores in the non-volatile memory inside the IC card chips. Figure 3 shows an example of a FeliCa non-volatile memory separation using AES as the cryptographic algorithm.

In the file system structure, system is the basic control unit in memory management. When Sony produces and releases the card, it only contains one System (named as System 0) that occupy the whole memory. Every time a new system needed to be built, it will run the system separation process, which will allocate enough memory space in ascending order in the system zero portion and named according to the chronological order. Each of the newly created system will occupy a unique memory location (no overlap with other systems) at system 0. In the actual operation of FeliCa IC card, the memory of different system are completely separated and will not cooperate. The design purpose of the system component aimed to provide multiple usage of the card in two or more completely different usages. This setting aimed to make use of the same physical carrier to handle multiple features with private running environment. So, in the reader's point of view, when presenting one FeliCa IC card with various system components in it, this is actually the case of presenting multiple logical cards to the reader. In this case, there are only one

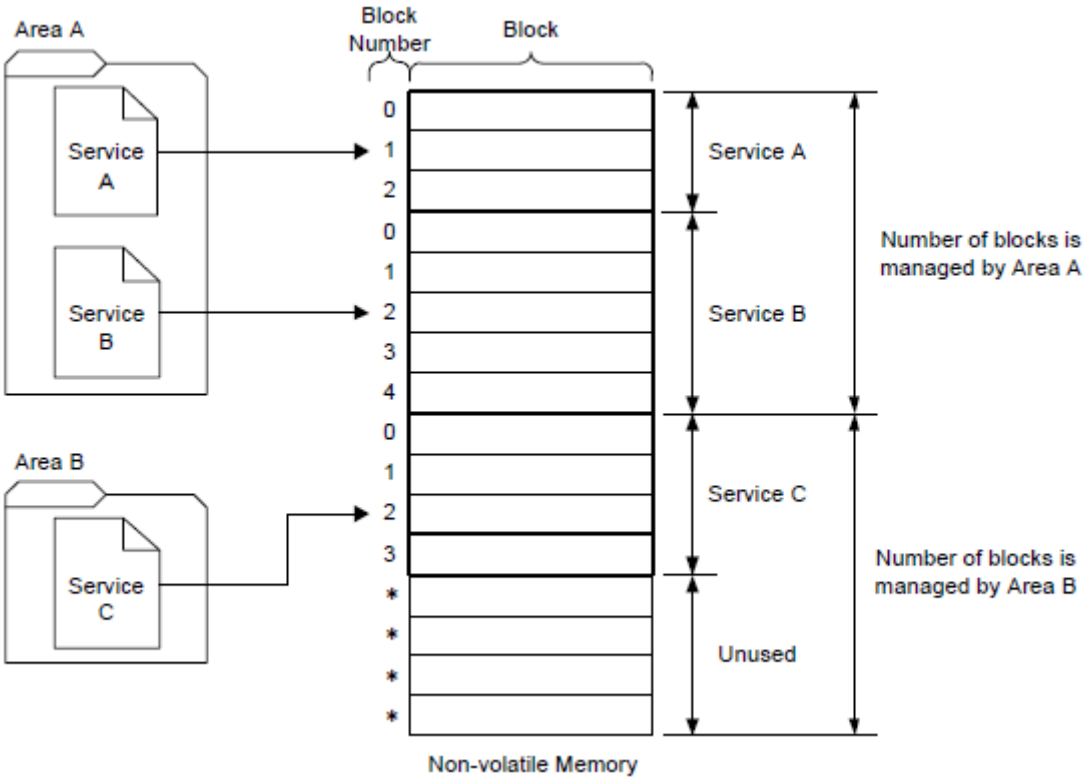

Figure 4: Structure of FeliCa memory with area and services separation

of them (or sometimes a few of them) is(are) the target to be controlled by this specific reader. All other logical card is no use to this specific reader. In other word, system component is just a logical separation of one physical card into multiple logical cards for use in different system usage. There are a unique key and id for different system that only known to the reader of the system. This setting can assure that the authorized reader of one system has no power to control the memory of the other system, which maintains the logical separation of the systems. When there are readers that are allowed to handle different system, it will need to issue a polling command to switch from the current system control to the others and will need a re-authentication. Figure 4 shows the structure of memory with concept of area and services.

Apart from complete separation of memory, sometimes there are services or applications need to share and cooperate with other services and share some of the memory access. And they may be used in a different area, like electronic purse and transport ticketing. Transport ticketing need to keep track of the travelling information. Sometimes it needs to check if the balance in the electronic purse is enough to pay for the journey fare, or need to deduct the amount from the electronic purse and keep track of how much has deducted for the electronic ticket. In this case, this two services need to cooperate. But they are services from two different areas and maybe provided by two different company. So, in this case, the two services will be located in a different area, and each area will manage their services and provide some limited cooperation between services in a different area. At the base of the file system hierarchy, each service will control their user data blocks which store data of the service. All the file system items store in the non-volatile memory. And each component (system, area, services) will have a separate key aimed for data protection and authentication when the components are set in an authentication required state.

In FeliCa file system, the service is classified into three groups. The first one is the random service, which is a general service provide read and write ability to control the data blocks. The second one is the cyclic service, which aimed to control the data block in a cyclic way which the newest data will be write to the data block contains the oldest data in the case of data block full. Otherwise, it will write in order in the data block. This service is mainly used for storing some tracking and logging information, like storing some log record of access time and date of access control usage of the FeliCa card. The last service is the purse service; it is intended to control the data block by increment or decrement action of the value stored in the data block. Also, it is possible to store some write once read many time data in data block managed by the purse service. The most prominent use of the purse service is the control of deduction and add value of the electronic purse. It is also possible to have multiple combined properties for a service, and we will name it as overlap service. All the data blocks will have a service attribute to mark that if the block is allowed to read / write / increment / decrement and also if the actions require authentication or not. These settings may be different for data blocks controlled by the same service.

2.3.3 Functions and Security Features

In FeliCa system, it also provides anti-collision features similar to the one used by MiFare system. This requirement is illustrated by the ISO14443-A standard which the two system follows. Apart from anti-collision features, FeliCa also provides anti-tear transaction feature which aimed to clean and discard all the uncommitted data and restore to a previous state for the last communication before doing anything before transactions. This function is provided in the case that some incomplete or corrupt data exists in the memory after a sudden removal of card power in the last transaction.

Encryption is also used for data exchange, communication and mutual authentication in FeliCa system. Unlike MiFare, FeliCa does not adopt a propriety algorithm for encryption. Instead, it adopts the conventional symmetric algorithm which is the 3DES and AES. There are three kinds of FeliCa IC card variants that apply different algorithm for data protection. They are the DES card, the AES card and the AES/DES card which support both type of cryptography algorithm and the actual algorithm used is based on the command used and the settings for data block access control. In order to increase the security level and prevent impersonation and replay, the actual keys use for encryption and decryption is generated at the time of use. It is done by some calculation on time, some particular value and the master key stored on both the control board and the IC card. This setting can help to provide a higher level of protection because the key used each time is different and so it is harder to recover the key or replay encrypted packages.

All these features and security measures provided by the FeliCa IC card increase the card security level. It also provides the FeliCa IC card a way to avoid many fraud situations and attacks. The high level of security of FeliCa IC card makes it the first contactless smart card brand to achieve EAL4 level in the ISO15408 standard [ISO09a, Pro99], which is the Common Criteria for Information Technology Security Evaluation standard. It aims to identify the security level of different information technology. The full specification of the FeliCa IC card can be found at [Son12].

2.4 Cipurse and Calypso

Apart from MiFare and FeliCa family, there exists two other famous alternative (standard) used in smart transport ticketing system. They are Cipurse[7] proposed by Open Standard for Public Transport (OPST) and Calypso[8] proposed by a group of European transit operators, including RATP and Innovatron in France. These two systems are rather new comparing to the leading standard of MiFare and FeliCa. They are mainly offered as an alternative to the two traditional smart transport ticketing system with more security and convenient features. But none of them has a great success to get a large market until now. And so the focus of this project will concentrate on the analysis of the protocol analysis of MiFare.

After a brief introduction of the smart card and some of the chip brand used in smart ticketing system. The security properties, weaknesses and possible adversaries of MiFare Classic will be introduced in the next chapter.

[7] http://www.osptalliance.org/the_standard
[8] http://www.calypsostandard.net/

3 Adversaries of Smart Transport Ticketing System

In the remaining part of this project, the authentication protocol of MiFare Classic will be used for illustrating the adversaries and attacks of smart transport ticketing system. Also, it will be used as the target protocol for evaluating the feasibility of symbolic adversary modelling on protocol analysis of smart transport ticketing system.

3.1 Authentication Protocol

The authentication protocol for MiFare Classic was meant to be a trade secret of NXP. It is kept as part of the proprietary algorithm for a long time. Because of this reason, the details of the whole authentication is always a research target for many researchers. In 2008, one of the research group disclosed the entire proprietary algorithm and loads of weaknesses of MiFare in [GdKGM+08], together with loads of weakness. Here is a brief summary of the authentication protocol that will be used as the target of protocol analysis by symbolic adversary modelling technique.

As mentioned in the previous chapter and [NXP11a, NXP11b]. The whole authentication of MiFare classic chip and the reader will include the anti-collision stage and the three-way mutual authentication stage. At the end of the anti-collision stage, the reader will only communicating will one tag chip and the uid of the chip will also be known to the reader, which will later be used in the mutual authentication stage. Figure 5 summarise the message exchange for the whole authentication.

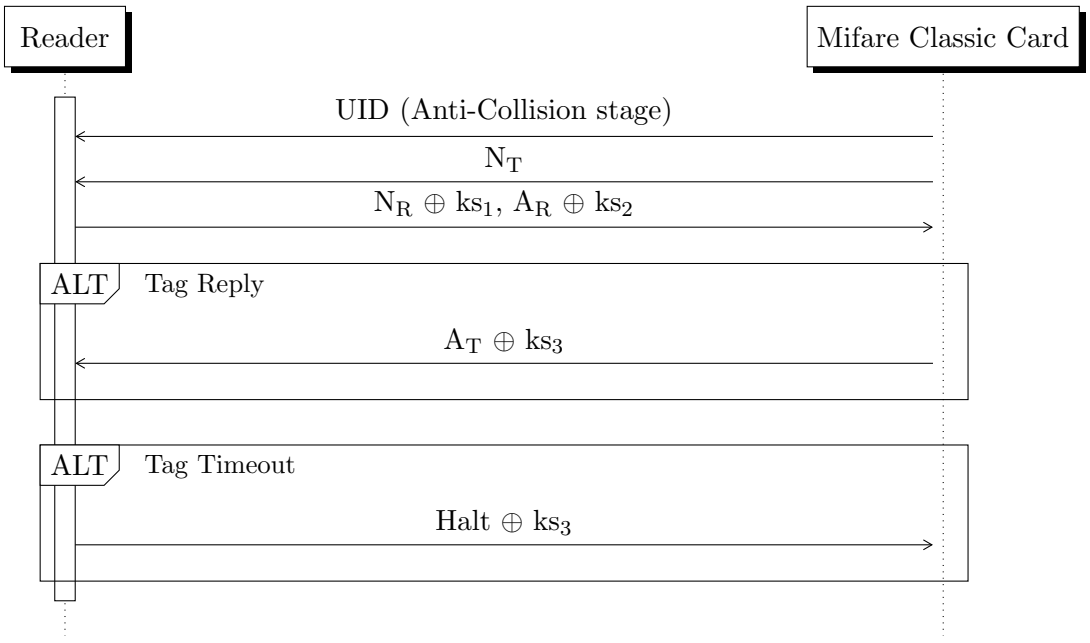

Figure 5: Mutual Authentication of MiFare Classic

3.1.1 Nonce and Response

It is discovered that, the two responses A_T and A_R are not calculated from the respective nonce N_T and N_R. Indeed, both responses are calculated from the nonce of the tag only.

In the tag and the reader side, there exists a 16-bits LFSR for the generation of N_T and N_R and the calculation of A_T and A_R. Each time when the tag is powered up, this LFSR will be initialized to the same known state. Then the LFSR will shifts in every clock cycle, providing an output bit (the left most bit of the current state) and a new bit. The left most bit will be discarded, all remaining bits will shift to the left, and a new bit will be added according to the current state by

$$NewBit = x_0 \oplus x_2 \oplus x_3 \oplus x_5$$

where the four input bits are coming from the old state. When the tag needed to generate N_T, it will take 32 output bits from 32 clock cycles of the LFSR and sent it as a nonce. When the reader received the 32 bits N_T, it will put the last 16 bits into a 16 bit LFSR as the initial state, then it will let the LFSR run for 32 clock cycles and discard all output bits. The reader will then take the next 32 bits as the response A_R, encrypt it and send to the card. Because the card also know the nonce, so it can calculate the same value and authenticate the reader. Then the tag will let the LFSR continue to run and get the next 32 bits as the response A_T and sent the encrypted A_T to the reader to authenticate itself. So, we can conclude that, if N_T is the first 32 output bits of the 16-bits LFSR, then A_R is the 65th to 96th output bits of the LFSR, and A_T is the 97th to 128th output bits of the LFSR. It is clear that, if an adversary eavesdrop on the communication and know the value of N_T, he can rebuild the LFSR and knows the value of A_R and A_T. So, in order to achieve mutual authentication, the encryption in the message exchange can only be done by the reader or the tag. If not, mutual authentication cannot be proved.

3.1.2 Crypto1 Encryption

Apart from the 16-bits LFSR, there are another 48-bits LFSR in the tag and the reader for generating keystream to encrypt A_R A_T N_R. As mentioned in the last chapter, each sector will have a key. That key will be the initial state of the 48-bits LFSR. The LFSR will keep generating keystream bit for the Crypto1 encryption. Each of the three target will be encrypted with a different keystream (ks_1 ks_2 ks_3), but they will all be generated by the same LFSR with the same initial state (the sector key). In order to add randomness to the keystream, the keystream generating LFSR will perform some xor operation with the nonce and the uid. If we know the value of sector key, UID and N_T (also the LFSR characteristics equation which is now reversed), we can use ks_1. Then we can get the value of ks_1 to get N_R and use it and the previous knowledge to get ks_2 and ks_3. The LFSR will shift at each clock cycle, but the output bit is different from a normal LFSR. Instead of taking the left most bit directly as the keystream bits, it uses some of the bits in the current state to perform some simple binary operations to get the result keystream bit. Those operations are all proprietary, but it has been reversed and mentioned in [GdKGM+08].

By just viewing the nonce, it seems that the nonce of the reader N_R has no use because both A_T and A_R depends solely on N_T with the support of a 16-bits LFSR. The fact is, N_R is used on the keystream generation for the encryption of A_R A_T N_R but not on the challenge response values directly. The action is done on the shifting of the LFSR at each clock cycle. For a normal LFSR, the updating of the state depends on the characteristics equation of the LFSR that will take some bits from the current state and xor them together to form the new bit and make it the last bit of the new state. In the LFSR used in Crypto1, it is a little bit different. Initially, the 48-bits sector key will directly used as the initial state of the LFSR. After that, the update of bit will first undergo the regular LFSR characteristics equation update, and then it will xor with the uid and N_T bit by bit for the first 32 clock cycles. After the first 32 clock cycles, where the bits for uid

and N_T is used up. It will do the same thing using N_R for the next 32 bits. This setting is where N_R is used for masking the state of the LFSR to make it harder for an adversary to recover the real state of the LFSR. It also hardens the discovery of the previous state of the LFSR by an enemy that eventually wants to know the initial state of the LFSR (that is the sector key and is the same every time). We assume that the adversary can eavesdrop on the communication and know N_T and finally calculate A_T and A_R. In this case, he can also recover the two keystream used in the encryption of A_T and A_R because it is just a simple stream cipher encryption with the xor operation. Also, in the case of RFID communication, there is always a chance that the tag leave the interrogate area of the reader and the tag will lose power and stop running suddenly. In this case, the card may fail to reply message to the reader in the three-way mutual authentication. If the reader does not receive the reply from reader, it will send out a "halt" command which is also encrypted by the next keystream. If an attacker can interrupt, discard or block the tag reply, he can also get the next keystream because the plaintext by the reader is always the "halt" command which is a publicly known constant. So by a simple xor with the constant value representing "halt" command, the attacker can retrieve that keystream. By also knowing the binary function (which has been reversed) on the LFSR state, it can restore the state of the LFSR. But the state of LFSR is masked by uid, N_T and N_R. So without the knowledge of the three value, it is hard for an adversary to recover the initial state of the LFSR which is the sector key. In reality, there are some weaknesses from this implementation and it will be illustrated in the following sections.

3.2 Security Properties, Adversaries and Attacks

When considering the security of the protocol, we always need to define the security properties that are required with the use of the protocol. Usually, the three big categories of security concern are defined by the three security properties of the CIA triad. These three properties will be briefly reviewed in the following session for the MiFare Classic system.

3.2.1 Confidentiality

By definition, confidentiality means to keep the data only accessible to those who have been authorized. In the case of MiFare Classic, the requirements are illustrated by the mutual authentication required before accessing data in different sectors of the memory. Each sector's access right are controlled by the keys and privilege settings stored in each sector trailer. The main security concern of the smart transport ticketing system in the confidentiality property will be privacy and secrecy of the card data. For every MiFare Classic chip, it is a passive RFID chip. So it does not control the start and halt of its operations. In the real situation, if we are implementing RFID chips with the thought of security. We will always need to control the security of the RFID chip by other means, like hardware control, mutual authentication or encryption algorithm. This setting is the main concern of the smart transport ticketing system as all the MiFare Classic chips and readers are operating in an open area. Any adversaries can easily eavesdrop on the communication between the reader and the tag, or using a fake reader or tag to gather information from the hardware for their future attacks. The adversary can violate the security rules by just eavesdropping on the communication channel or the hardware.

For some of the MiFare Classic card used in smart transport ticketing system, the card itself will store some sensitive information like the balance or some travel habit informations. If the card is personalized, it may even contain more personal information of the

card holder. Potential adversaries can get hold of those information and violate the privacy of the user. After they get that information, two possible attacks are tag tracking and tag(card) impersonation. For tag tracking, the adversaries may get hold of the user location information, or even the travel habits from the card used for travel. For example, an adversary can just sit next to the victim in an underground station and use a fake reader to get hold of the informations from the smart card, including the entrance and exit records, money remaining and other payment and travel habits. These kind of informations can be sold or used in other places, or even act as a source of other attacks like social engineering. It can also help robbers to identify people with more money that can be indicating by a high level of remaining value in their smart card. It completely violate the privacy of the cardholder in a way unattended to the user. Apart from tag tracking, another attack base on violation of confidentiality is the tag or card impersonation (card cloning). The adversaries can eavesdrop on the communication channel and get hold of enough informations to impersonate an actual card. In this case, if an adversary can reverse the sector key from a genuine tag or able to reply correctly to an honest reader, then the impersonation is success, and the adversary can do whatever they want to pretend to be a legitimate user. Including a free ride on transportation or take the transportation as the identity of a legitimate card holder. The possible attacks and adversaries for confidentiality property can solve by a strong mutual authentication between the tags and the readers. Unfortunately, the mutual authentication used by MiFare Classic based on the Crypto1 encryption scheme is not good enough to confirm that only the legitimate reader and tag can complete the mutual authentication. Also, some real implementation of the MiFare Classic has some implementation flaw which lead to a broken authentication. It will be mentioned later in this section.

3.2.2 Integrity

Integrity is another important property for the system. It aimed to keep the data unmodified by unauthorized person. This property can be targeted on the message exchange at the communication or the credential information stored in the card memory. Considering the case of the smart transport ticketing system. The message exchange can include the nonce and answer sent during the mutual authentication stage, or the information passed for write and read on the memory of the smart ticket. The data or credential information stored in the card memory can include the money balance, owner details or even the key and access privilege for controlling the access of the memory sectors. The main concern on the integrity of a smart transport ticketing system will be card data modification, replay and relay attack result in card (or user) impersonation.

In the smart transport ticketing, some important data are stored on the card. These data may include the valid period of a yearly or monthly ticket, or the cash balance of the account linked to the smart ticket. This kind of information is directly related to the payment details or the ticket validity of the card holder. If these kinds of information does not have integrity protection, then there maybe some enemies change those information and enjoy a free ride on the transportation and totally violate the implementation of smart transport ticket. This problem is one of the main reason for implementing mutual authentication and access control before allowing parties to read and write on card memory. The weaknesses on the authentication will also allow unauthorized data modification on the card memory. Apart from the advantages taken from the illegal data change, adversaries can also violate the integrity by illegally using the identity of a valid card or pretending to be a real card. Adversaries can use replay and relay attack to do a card (or user) impersonation. They first eavesdrop on the honest communication between the

legitimate reader and card and try to use the information retrieved to start a new section. The adversary will reuse those captured information to complete the authentication or communication with the reader (or card) in order to let the coupling device think that it is a legal device. The only different is for relay attack; the enemy is standing between the actual card and reader and relay all message between the communication channels and impersonates the coupling device in real time. And replay attack is done in asynchronous timing, the adversary may imitate the genuine identity later to gain advantage to the service. Both of the attack is using a weakness of failing to check a real freshness of the nonce.

3.2.3 Availability

Availability property aimed to ensure the system for providing service for the intended users in expected time. This requirement is also an important factor for smart transport ticketing system. From the time when the smart transport ticketing system is implemented to replace the traditional ticket, there are lesser need for a ticket inspection by real people. This also mean that, if the automated smart ticket inspection is not available, it will cause a lot of trouble because many of the stations rely solely on the automatic system. Also, considering the case that a user just have a smart ticket (like Oyster Card) with no cash in their pocket. Then it will be a mess to this user if all the automated smart ticket inspection is not available. He will not able to have a valid ticket on the journey. So keeping those systems available is one of the security concern of smart transport ticketing system. Possible adversaries and attacks in this category mainly aimed to interrupt the communications between the tag and the reader by either passive or active actions. They can be some physical barrier, jamming or interference on the communication channel, or purposely disable or kill a tag or reader.

3.2.4 Weakness on Authentication Protocol

Originally, the authentication and all its working is proprietary and are the private properties of NXP. No one knows that is it safe or not because it aimed to provide security by obscurity. In a later time, some of the researchers dismantle the protocol and find loads of weaknesses on the Crypto1 algorithm, this discovery let the assumption that all security is relied on the authentication protocol become invalid. Here is some of the weaknesses discovered on the authentication protocol.

As mentioned above, all the expected response from tag and reader used for identity proving in the mutual authentication can be deduced by using a simple LFSR. So, the security protection on the identity proving in the mutual authentication solely depends on the encryption that requires that the generation of keystream are variable. The keystream used in the Crypto1 encryption is created by LFSR with known characteristics equation. And so the only thing kept secret is the initial state of the LFSR that is a key stored in each memory sector trailer for MiFare Classic card. If there are weakness on the authentication protocol or the design on the Crypto1 LFSR, there is a possibility to leak out the value of the key. This flaw will let the enemy knows the initial state of the LFSR and can generate all keystream. Because the adversary can get hold of some keystream, so we can conclude that the adversary can recover the state of the LFSR at the time point. Thus, the design of the LFSR must able to stop the adversary to reverse the action of the LFSR and find the previous state. If this action is possible, then the adversary can repeat the operation until reaching the initial state and get hold of the key and are able to generate any keystream. As a result, the adversary can successfully pass an authentication by knowing the sector

key. In reality, there are many mathematical weaknesses that can help the adversary to recover the sector key. Summaries can be found at [dKGHG08, Tan09]. Another important consideration is, the Crypto1 LFSR is masked with the two nonce and the uid of the tag when it is shifting. In this case, it is also an excellent way for an adversary to recover the previous states of the LFSR by finding those three values. In reality, the nonce from tag and uid is transfer in clear, and we always assume that the adversary can eavesdrop on the communication channel and get hold of those information. So, in order to make the reverse of LFSR state harder, the value of the reader nonce (which is the only unknown value for the enemy based on his initial knowledge) need to be highly unpredictable and also not repeating for a long time. Unfortunately, the LFSR generating the reader nonce is just a 16-bits LFSR. By definition of LFSR, it will only able to produce 2^n-1 different values before repeating the cycle where n is the number of bits of the LFSR. That is 65535 in the case of Crypto1. It will repeat in a very short time and because the clock cycle time of the reader is fixed. It is quite easy for an adversary to predict the value of the reader nonce. Or the adversary can even wait for a same encrypted nonce and replay the answer without knowing the real nonce value. This flaw makes the authentication broken as it has no confirmation that the device replying with a response is from the legitimate card.

3.2.5 Weakness on Implementation

The MiFare Classic is made and sold by NXP. They need to test the card before selling to the user of some MiFare implemented system. In order to test the authentication and memory access of the card, there will be a need to include sectors key inside the card memory sector trailer. For this reason, after the manufacturing of the card, there will be some default key stored in the card memory sector trailer for testing. It will also be used when the real system implementers needed to test the card on their system. By the official documentation of NXP, those default key value has been listed, and the system implementers are suggested to change those key value before putting them into actual use. But the reality is, not all of the implementers aware of this and they put the card directly into use without changing the default keys. As mentioned above, the entire security concept of MiFare Classic primarily depends on the secrecy of the sector keys only. If it is a fixed known value, the whole security concept of the MiFare Classic collapses. So, this can be considered as an implementation weakness of the MiFare Classic. Apart from this, for some basic system implementation, there will be many cards (tag) active around the system. If each of them requires a unique sector key, then the reader will need to keep track of the mapping of keys to all card sectors. This setting will be very memory and time consuming when the active cards are increasing. So in some of the case, all the card will share the same key for the same sectors. This is another potential weakness on the implementation of the MiFare Classic. If an adversary can dismantle and know the key of one card, then all the active card is in danger.

Besides possible leakage of sector keys by poor implementation, there exist another implementation flaw on the system. As the MiFare Classic tag is passive RFID tag. So it will only powered on when it is in the interrogating field of the reader. When the tag is out of that area, it will lost the power, and no status can store in that case. By this understanding, it is known that each time when the card power on, all of its LFSR will initialize from a known state. So, with this knowledge and careful handling, an adversary can control or know the nonce value of the tag by waiting for a suitable time for the LFSR shift clock cycle. This method makes the adversary get knowledge of the sector key easily because the nonce of tag is fixed and different in the encrypted response can be additional information for an attacker to do the reverse on the

Crypto1 LFSR to recover the sector keys. More weaknesses and attacks can be found at [dKGHG08, Tan09, SVRG+08, GdKGM+08, Cou09, GvRVS09].

3.3 Improved Protocol

From the brief description of the original MiFare Classic mutual authentication, and also some weaknesses and possible attacks. We can conclude that the original MiFare Classic fail to retain some security properties that are required for smart transport ticketing system. For this reason, some of the researchers have proposed some improved mutual authentication protocol to be used on the system implemented by MiFare Classic. It is worth mentioned in this section for two of the improved protocol because they can also be a target for symbolic adversary modelling and the evaluation in the later chapters.

One of the weaknesses of the original MiFare Classic mutual authentication protocol is that every time the 48 bits LFSR used for keystream generation start from the same state, so one of the improvement is to add randomness to the initial state for the generator. This method can help to decrease the chance of replay attack because the keystream used to encrypt the nonce and response is different each time. It is harder for an adversary to predict those values comparing to the original authentication which will reset to the same initial state every time a tag power on. This improvement is proposed in [HC12] and is illustrated in figure 6 below. For this proposed authentication protocol, ks_1 and ks_2 are generated with the knowledge of the LFSR, the sector key K, UID and the time stamp. Keystream ks_3 and ks_4 and all other keystream used for encryption later in the communication are deduced with the knowledge of the LFSR and the two nonce N_R N_T. So the third and fourth step can be treated as secret value exchange for the keystream generation of the message exchange onwards.

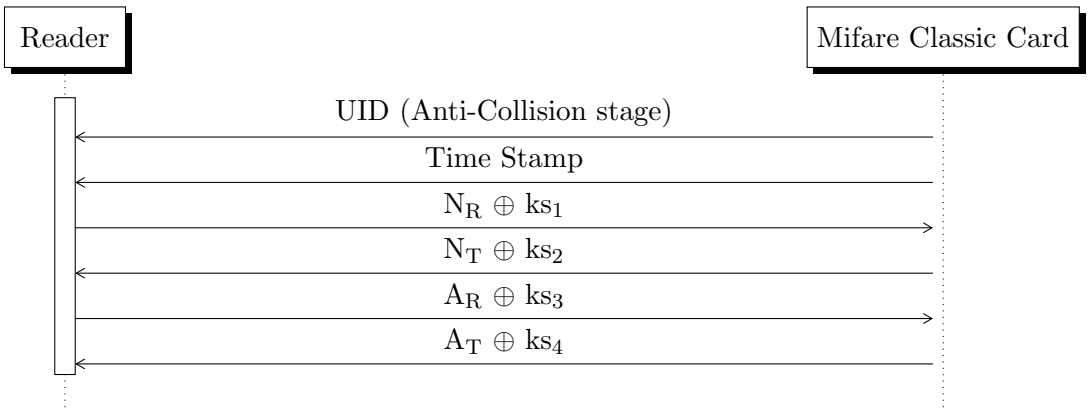

Figure 6: Proposed Mutual Authentication of MiFare Classic in [HC12]

There is another improved protocol proposed in [JK13] which aimed to solve more security concern for original MiFare Classic mutual authentication protocol, including some defence on replay and relay attack. Also, it aimed to provide more protection on the reversing of the 48-bits LFSR used for Crypto1 keystream generation. The proposed protocol is illustrated in figure 7 below. In the authentication, the card will first send a nonce N_{T1}, and both the reader and the card will deduce ks_1 as same as the way use in the original Crypto1 algorithm. This method is making use of the shared sector key, UID and the first tag nonce N_{T1}. In order to add randomness to the keystream, a S-Box will also be used. It will take some input that are only known to the reader and card (possibly the other sector

key or the xor result of both sector keys) and output some bit to xor with the keystream to add some obscurity. This method makes it almost impossible to reverse the state of the LFSR because a non-linear S-Box is added during the process to mask the keystream, which is the output of the LFSR. With the first keystream and the 32-bit output from the S-Box, the reader will then create two nonce. The first nonce will be xor with the first keystream and the output of the S-Box to get the cipher text. Then the first nonce will xor with the next 32-bit of output from the S-Box (the last output of the S-Box will serve as the input of S-Box this time) and become the initial state of an LFSR to produce the next keystream. Then the second nonce will do xor operation with the second keystream together with the next 32-bit output of the S-Box (still using the last S-Box output as the input) to get the second ciphertext. The tag will do the same generation and validate the structure of the S-Box value xor with the keystream and get the first nonce N_{R1}. Then it will use the first nonce as an additional knowledge (for generating ks_2) to retrieve the second nonce N_{R2}. The similar thing will be done again in order for the card to generate N_{T2} and for the reader to verify it. In this protocol, the authentication works because create of ks_1 will request knowledge of one of the sector key and the create of R_1 will need knowledge of the other sector key. Also, it requests the knowledge of the first card nonce N_{T1}. So, the first part of message 3 can act as a response to N_{T1} if the tag can validate the output value of the S-Box and ks_1. The second part of message 3 will serve as a challenge from the reader to the tag, and it work the similar way which the tag (card) reply it on message 4.

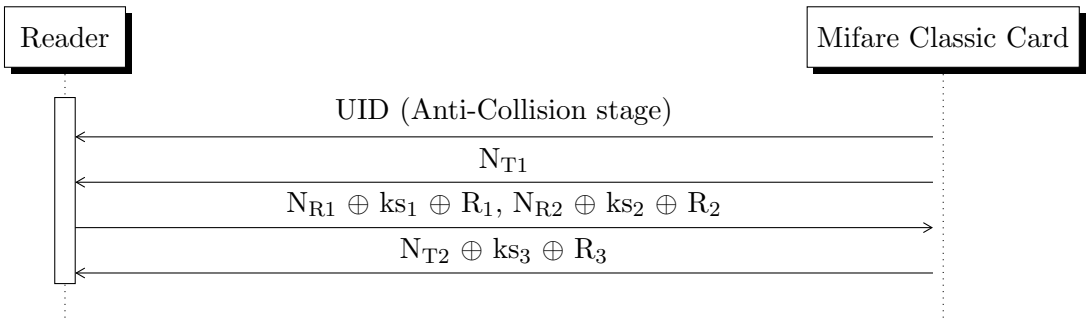

Figure 7: Proposed Mutual Authentication of MiFare Classic in [JK13]

After introducing the possible adversaries and two improved authentication protocols of the Smart Transport Ticketing System. Some existing adversary modelling approach will be introduced in the next chapter.

4 Adversary Modelling

Adversary model is a kind of modelling technique uses to review the security level of the system, protocol or other logical settings with a set of assumptions and estimation of the computation power, initial knowledge and ability of the possible enemies (can be active or passive attackers). Sometimes, it can help to discover potential problems in the design and reduce the risk by fixing it before the plan is implemented in the real world. There are many different ways to model the security level of a system or design, including threat modelling, which aimed to examine the goal of the potential adversary to minimize the effect of the adversary on the system. Another way is attack modelling, which is aimed to analysis not only the goal of the adversaries, but the attack path of the adversaries. For example, the privilege escalation steps of the adversaries on the system or design. Lastly, protocol modelling, which is the main concern of this project and aimed to analysis the message exchange and the communication channels. It targets to analysis if the information leak to the adversaries in the communication process causes security problems to the system and design. Examples, brief introduction and comparison of existing adversary models can be found in [BAN90, Mea91, DM99].

4.1 Symbolic Adversary Modelling

In the different kind of adversary modelling technique, symbolic adversary modelling is preliminary aimed to do protocol analysis, which suits the need of this project that aimed to analysis the feasibility of adversary modelling on smart transport ticketing system protocol. For symbolic adversary modelling, it mainly makes use of some mathematics symbol and notation to define the communication and the settings of the protocol. Then try to use math approach, like set theory to analyse if the protocol is secure provides with some assumptions, mathematical proves and deduction. In some of the symbolic adversary model, it will define a new set of protocol algebra language to represent each of the communications and knowledge in order to simplify the analysis process and the complexity of the modelling work. It can also help to provide prove that the simplified version is an analogue to the full protocol. Some of the symbolic adversary model are very famous and so some of the researchers has even developed some automatic tools to assist in the modelling and analysis process. Some existing symbolic adversary modelling framework will be briefly introduced in the following sections. More knowledge on Symbolic Adversary Modelling and Computational Models can be found in [Bla12].

4.2 Process Algebra CSP Model

Communication Sequential Processes, or CSP in short is a mathematical and symbolic modelling framework, which intended to use a mathematical approach (mainly in set theory) and a defined set of process algebra symbol to study the concurrency and communication between the honest communicating parties. In the case of smart transport ticketing system, the modelling model will be based on the communication sequence between the readers and the transponders (that is the chip in the card). Also, the behaviour of the intruders and adversaries who are also exist on the communication channel will be modelled and analysed together with the legitimate parties. The concept of CSP is first introduced by Professor Hoare in [Hoa85]. And the following subsections aimed to review the basic CSP model and the extension of it that is done by many researchers and authors

after the Professor Hoare's proposition in his book. There are one thing to notify that, in CSP modelling and analysis, it will only concentrate on the communication process, and are not suitable for analyse on the security level of a cryptographic algorithm design. But there exist some possibility to extend the initial knowledge set of the adversaries if he can deduce something more from the original knowledge because of a flaw in the cryptographic algorithm, say Crypto1 used in MiFare Classic.

4.2.1 Process Algebra Language

In the book of Professor Hoare [Hoa85], he proposed a set of process algebra language and symbol (that is more or like a variant of the symbol using in set theory) designated for CSP modelling framework in order to provide much simplicity in the modelling process. This introduction only acts as an insight on the CSP modelling and the linkage of adversary modelling. It only includes base modelling by mean of the word and symbol, but do not provide a full range of determining method to analyse and illustrate that the communication is secure or not. In other word, Professor Hoare just introduces a symbolic method to represent the message exchange in a communication sequence between parties. It does not provide ways to review the model or protocol after it has been illustrated by the CSP language. Prof. Schneider is the first researcher to mention about how security properties of a protocol or communication sequence modelled by CSP language can be defined and analysis. The proposition of Prof. Schneider can be found at [Sch96]. After that, there are many researches aimed to extend the CSP modelling framework to be a full system of symbolic adversary modelling framework. They aimed to provide different kind of security properties proving and possible loophole illustration in the communication illustrated by the CSP language, which is a similar direction done by Prof. Schneider. They also suggest many extensions to the original symbol and language set of CSP. There are two books mentioned and cited by many researchers and authors, including another book from Prof. Schneider [Sch99], which intended to give details on discrete-time modelling. And the book from Prof. Roscoe [Ros97], which cover a large scale of an extension on the basic CSP modelling for time unrelated and insensitive communication.

Base on the mathematical model of a communication illustrated by the CSP language, many researchers proposed ways to analysis, test and illustrate the security level and possibly attack of the protocol. The analysis of the CSP model can be separated into two ways, the model-checking approach and the model theorem-proving approach. This two approaches go deep into the symbolic model and provide outcome and illustration of the needed security level and loophole of the system under predefined set of abilities and initial knowledges of an adversary. In the model-checking approach, there are an automated tool to complete the checking. The theorem-proving approach makes use of the Rank Function, which is first introduced by Prof. Schneider in [Sch98] and aimed to check if information is leaking in the communication channel which increase the space of the adversary's knowledge. The following subsections will review the signalling of CSP and the two approach in protocol analysis.

4.2.2 Signalling

CSP is targeted to model the possible behaviour of adversaries and honest communication parties in the communication process with the symbolic interpretation and the checking of the message and communication sequence sent in the channels. In some of the case, only part of the security features of a protocol wanted to be modelled because the pro-

tocol is just created to have some of the security features which the other part is not necessary. In other word, the protocol may design for one security goal only. So, the security features that are needed to be analysed should also be defined in the CSP model of the protocol in order to study it with the right approach of required security features and without too much concern on other unimportant security features that are not the target of the protocol. In this case, CSP modelling provides a way to specify the security target and features that are required to analyse. That is the signalling message in the model. The main idea is, the communicating parties will send out signal as part of the message modelled in the CSP model. The signal is intended to show and declare that at that precise point of time, that party should confirm that the protocol has achieved the designated level of security. For example, if A and B is communicating in a protocol, and after a simple challenge-response authentication, A can send a signal indicate that at that time point, B has authenticated to A. B provides sufficient information to prove that he processes the secret key shared between A and B. In this situation, the security requirement is stated in the model indicating if the security level of the authenticating protocol is secured under the present of adversary, then A could send out that message to indicate the success of authentication. The two checking approach (model-checking and theorem-proving) is aimed to find that if A send out that signal of claim, is there any ways or points in the protocol will leak out information to let the claim become a false claim. If there exist a message in the model that violate the claim, then the protocol is insecure. The signalling provides a method to identify what security measures will need to study in the CSP model defined for the protocol. There is a detailed illustration of signalling in [RSG+00].

4.2.3 FDR and Casper

FDR is an automated commercial tool[9] [For03, Low96] which intended to go deep and walk through the trace of the model in order to check for potential problems under adversaries. The tool is broadly used for the model checking approach of CSP model analyse. It mainly checks the specification and implementation set of the protocol that is modelled by CSP language. Then it proposed to find that if the implementation meets all the constraint set in the specification of the protocol. In other words, it aimed to check if the implementation set is the possible refinement for the specification that the potential enemies gain nothing useful from the message sent in the communication channels. If the result is false, then the FDR can provide a counterexample that is a trace of the route to reach a state of possible specification violation. If the result is true, it is reasonable to say that the implementation fully achieved the specification, and there should no trace in the implementation that will violate the security level set in the specification.

In the CSP modelling framework, the most important part is to read the details of the protocol design and to write out the symbolic format of the message exchange and all actors' behaviour in the communication channel. In the real situation, this part will be very time consuming because many protocol is gigantic and include a lot of message exchange even after simplification. Also, errors are easily found in the creation of CSP model for big protocol that contains lots of mathematical and algebra symbols. In this case, it will affect the correctness of the analysis. For example, a single specification error in the FDR input can lead to a false positive case. Because of this reason, G. Lowe [Low98] has developed a compiler like software named Casper to help to transform easy understand wording to the CSP model in mathematical symbols. This method is very similar to the compiler to

[9] A product of Formal Systems (Europe) Limited

compile high-level language like C or Java to the low-level language of assembly and byte code which need precise control and systematic process. The compiler will provide CSP model as output, which can be used in rank function theorem-proving approach or act as an input to the FDR for model checking approach.

4.2.4 Rank Function

For the theorem proving approach of the CSP model analysis, the most-used one is the rank function, which is first proposed by Prof. Schneider in [Sch98] and refined in [SD05]. It intended to give a rank for each of the message transmitted in the communication channel (which is the CSP model for the protocol) to find that if possible message will leak out information to the adversaries and increase the knowledge base of the enemies. It provides a general proof to tell that if the CSP model and all its possible trace is in the expected security level. The main idea is, each message exchanged in the communication and the initial knowledge of the opponents will be given a rank (an integer). In other word, the rank function provides a mapping between the facts / signals / message and an integer. The communication parties will need to maintain positive rank in the communication. That is if a party only receive a message with positive rank, then it can only send message of positive rank, with the exception of the adversaries that are only allowed to send positive rank based on the initial knowledge of himself and all the message send in public. All secret values like the shared secret key of the MiFare Classic card will carry a non-positive rank. If the communication parties just sending positive rank message in the channels, the parties can only receive positive rank message which means that the secret or signal claim will never appear on the channel that could increase the knowledge of the adversaries. The rank theorem proving aimed to find a set of rank function that fulfil all the requirement of the rank function. If there exists such a rank function for the protocol in all of the considerable cases. Then the protocol is proven to be secure on the designed security features which mean that, in all case, it can be proved that if the user send out a signal with security claim, then the claimed items must not leak to the adversaries. And if such information has leaked to the adversary, then the user will never get a chance (by rank function maintaining positive rank message) to sent the claim. Thus, the leakage is notified and will not violate the rule that a security claim will only announced in the case of security requirement met. In [Hea00], it describes a way to find that if a rank function exist for a given CSP model.

4.3 Dolev-Yao Model

Dolev-Yao model is one of the famous and widely spread symbolic adversary model. It is first introduced by Danny Dolev and Andrew Yao in [DY83] and aimed to analysis the public key cryptography protocol. It uses a newly defined set of protocol algebra to describe the protocol communication, message exchange and the principle which may include the initiators, responders and opponents. It tries to monitor the behaviour of the adversaries in the protocol communication by a set of assumption. In the assumptions, an active adversary can perform action supported by the protocols to create and obtain information of the protocol. But the adversaries are not controlled on what message he could send in the communication environment. He can send any message he likes on the protocol for his interest. As a result, the model can represent and interpret the possible behaviour of the opponents and the honest communication parties and start an analysis base on those assumptions and protocol algebra model.

Originally, the Dolev-Yao model proposes in [DY83] mainly targeted on public key cryptography. So the set of protocol algebra introduced in the paper is just enough for public key cryptography. Some researchers have tried to extend this set of symbols to support more kinds of protocol analysis, like [Sel03, BC10, HP03]. The set of symbols defined in the model bases on part of the symbol used in set theory, which make it easier to understand without the need to learn a complete set of new modelling language. It also provides an easy way for researchers to extend the model to study protocol besides of the public key cryptography protocol. Indeed, the first variants appear is a simple extension of the model to use on the adversaries modelling on symmetric encryption protocol.

In the base of Dolev-Yao model, it aimed to analyse the message exchange in the communication channel between honest communicating parties with the present of adversaries. This assumption is the basic settings of the Dolev-Yao model. From this setting, it also provides a few set of assumptions in order to make the protocol analysis simpler in nature. Firstly, it assumes that the original communicating parties are honest and will strictly follow all the steps and requirements of the protocol. For example, in public key cryptography, the protocol requires the communicating parties to keep their secret key only to themselves and should not release it to any other parties. Secondly, the adversaries exist on the communication channel can do anything and have a full control over the channel and on the channel only; they will still need to follow the functions that are not in the channel. For example, the adversaries can eavesdrop, replay, relay or even delay the messages appear on the channel, but they cannot tear apart a message and read the content if the decryption is only possible by a key that is not in the procession of the adversaries. There are also an additional assumption that the modelling will just concern on the message exchange in the communication channel, and it will treat all the functions and cryptographic measures as a black box and paying zero interest on the flaw happen on those part. These kinds of assumptions make the modelling and analysis of Dolev-Yao easier to use.

After a general introduction of the two existing adversary modelling approach, the base of CSP modelling and the new approach extending from CSP will be illustrated in the next chapter.

5 Adversary Modelling on Smart Transport Ticketing System

In this section, the base of CSP modelling language, with the theorem proving scheme of the rank function will be illustrated and analysed. It will aim to follow the path of Prof. Schneider illustrates in [Sch96, Sch98] with some additional items to consider the change of the adversary knowledge during the communication sequence. At the end of this chapter, the combined approach with the CSP based modelling and the additional consideration on adversaries' changing capacity will be concluded. In the next chapter, the proposed approach will be implemented in the adversary modelling on the MiFare Classic authentication protocol and the improved protocol. The suitability of the proposed approach will also be analysed and evaluated.

5.1 CSP Modelling Language

In [Hoa85], CSP modelling language is introduced as an abstract language to model all the communication sequences, events and processes happen in the protocol or system. All message exchange and activities done between actors will be represented by this set of language. After the whole system has been modelled by the CSP language, there may exist more than one possibility of the messages or actions flow. Each of them is a trace to the system behaviour. By analysing the trace, with the help of security properties signalling and rank function theorem proving, the security properties of the communication protocol or system can be determined. These basic CSP modelling approach will be defined in this section.

5.1.1 Processes and Events

The base of CSP modelling language is to use an abstract formal language to define all the behaviour and action of actors exist in the system or protocol. The most-important information for communication is a message exchange between actors, which is shown as following.

$$Send.A.B.m$$

There are four part in this representation, and a dot separates each part. The first part means the action done by the actor mentioned in the second part, and the target actor of the action is shown in the third part. The last part of it represents the content of the action. So, this statement can be read as actor A send message m to actor B. In the viewpoint of the adversaries, it is sometimes not necessary to know the direction of action, it just need to use the message in the communication channel for his own good. So the first three part of this statement can be considered as a channel and it can interpret as there is a message m appear in the communication channel of Send.A.B which involve any actor that have access to this channel, include A, B and all possible on path adversaries.

The status and knowledge of the actors are keep changing in the communication process. The legitimate actor may set up a new secret key with another legitimate actor. Or they can authenticate the identity of the other actor. Also, the adversaries may gain knowledge from the channel to increase his knowledge. In the CSP modelling approach, the state of a process or actor will always behave differently after the event occurs. The following statement means that you can perform an event "a" provided by process P and

after a is performed, the process will behave likes P.

$$a \rightarrow P$$

P can either be the process that the actor will perform or the change in status of action. If that event is the last event of the communication, the process will stop running after it. And so the following means that if a message m is passing through the channel of Send.A.B, that is if a message m is sent from A to B, then the process will stop (indicating by the STOP process) afterwards.

$$Send.A.B.m \rightarrow STOP$$

There is also a representation in the CSP approach regard to the two special communication channel of input and output. The following statement means that an actor will accept and input k from the channel c and then behave like P(k).

$$c?k \rightarrow P(k)$$

For the process P(k), it represents a function of the process P after receiving the input k. If the process represents an actor of adversary, it means an increase in the adversary knowledge. In the following, it means that when an adversary see a value k in the channel c (or there is an input k send to the adversary from channel c), it can accept the value k and use it to increase its original knowledge set K and become a new state.

$$c?k \rightarrow Adversary(K \cup k)$$

Apart from the input, the following statement means a process will behave like P after it has outputted a value k in the channel c. It is a similar usage as input, with a different symbol.

$$c!k \rightarrow P$$

In a typical situation, the communication will always have some alternative detours. There will be some choice of action for an actor to choose in some points in the communication. In order to model this kind of situation, CSP modelling language provides a choice operator with two different kind of usage. The following statement assume that P(i) is a limited or unlimited set of process event with index i to represent any possible alternative. For example, if set I equals to {1,2,3} and P(1)/P(2)/P(3) refer to Send.A.B.m1, Send.A.B.m2 and Send.A.B.m3 respectively. Then the following statement mean that there is a choice between Send.A.B.m1, Send.A.B.m2 and Send.A.B.m3, when any of the choices is performed, the process will stop.

$$\Box_i P(i) \rightarrow STOP \qquad where \;\; i \in I$$

The above representation is used when a set of choices are large or with uncertain size. If the choice has a limited options, it can be shown as follow, which means that either Send.A.B.m1 or Send.A.B.m2 is performed, the process will stop.

$$Send.A.B.m1 \;\Box\; Send.A.B.m2 \rightarrow STOP$$

Concurrency is also an important factor in the communication process. The following statement means that the process P and Q run independently on all cases, with the exception of events in the set A which they need to run synchronized.

$$P \underset{A}{\parallel} Q$$

There is a particular representation which a process is parallel to the stop process in some event. Because the stop process means the end of the communication and can do nothing more, so if the process is parallel with the stop process in the event a, it means that the process will stop on event a. The meaning of the following statement is process P will keep running and will stop on event a.

$$P \underset{a}{\parallel} STOP$$

The last set of symbol in the basic CSP language is used in the interleaving case. The following statement is very similar to the choice operator. The only different is, with the choice operator, the process will stop if any of the processes in set P(i) is performed. In the case of interleaving operator, it means that all the process in P(i) will perform freely with any order and any concurrency status, and then it will stop.

$$\Big|\Big|\Big|_i P(i) \rightarrow STOP \qquad where \;\; i \in I$$

Also, similar to the case for the choice operator, there exists another way to present cases with two process only, they are shown as following.

$$Send.A.B.m1 \;|||\; Send.A.B.m2 \rightarrow STOP$$

5.1.2 Actors, Systems and Network Environment

In the last section, the basic syntax of the CSP modelling language has been introduced. Usually, each statement in the modelling process will consider about one actor or system only. It is different from the reality that there will be at least three communicating parties in the entire communication sequence. They include the two genuine communicating parties and an additional adversary. In order to model the whole network environment and to study all possible participating actors in the communication sequence, we will model the system by considering the entire system. Then each of the possible actors will be modelled separately. In a network with two legitimate actors A and B with an adversary C, the network model represented by CSP model is shown as following.

$$NETWORK = (A \;|||\; B) \underset{send,receive}{\parallel} C$$

In the above statement, it means that in this network environment, there exist only three parties. All the three parties can run parallel in the environment, and each of the legal actors will synchronize on the send and receive channel with the adversary, and the actors themselves can run independently in any order. In this network, it means that A and B will start a message exchange between them, and the adversary can creates or eavesdrops on those messages exist on the send and receive channel. The behaviour of each actor will be analysed separately by defining the behaviour of each action using the set of basic CSP modelling symbol. In some environment, there are unknown or large number of potential users. Any pairs of user will start a message exchange randomly. For example, in the case of MiFare Classic used in Oyster card system. The group of actors includes all the readers installed at the ticket barrier in each station, together with all the oyster card sold to the customers. You can never predict which card will authenticate at which reader in any time. In this case, the modelling of network can be described as follow.

$$NETWORK = (\Big|\Big|\Big|_{i \in U} USER_i) \underset{send,receive}{\parallel} C$$

The statement above defined a set of all possible user U and the set of all possible user behaviour USER. In this case, the behaviour of each user i is described in User_i and will be analysed separately. The following is a full modelling of a system with an arbitrary user A send a message m to another user B, where B and the adversary can obtain it. The adversary then uses this message m to increase its knowledge set K.

$$NETWORK = (|||_{i \in U} USER_i) \underset{send,receive}{\|} C$$

$$User_A = \Box_{b \in U} Send.A.b.m \rightarrow STOP$$

$$User_B = \Box_{a \in U} Receive.B.a.m \rightarrow STOP$$

$$Adversary(K) =$$
$$\forall_{i,j \in U,m} \bullet (Send.i.j.m \rightarrow Adversary(K \cup m) \,\Box\, Receive.i.j.m \rightarrow Adversary(K \cup m))$$

In the second statement above, the choice operator of any b in user set U means that at the time when a random user A send out a message, he never knows who will receive it because the receiver has not yet authenticate. So it can be received by any user b in the user set exist in the network environment. This setting is the same case for the third statement that an arbitrary user B receive a message that the sender is uncertain.

5.1.3 Trace Logic

In the basic syntax illustrated above, it is possible that many detours exist in the message communication. This phenomenon is modelled by the choice operator in the basic CSP language. In this case, there exist a lot of paths that can reach the final state in the statement. For example, the following statement can have four paths before the process stop.

$$(Send.A.B.m1 \,\Box\, Send.A.B.m2) \rightarrow (Receive.A.B.m3 \,\Box\, Receive.A.B.m4) \rightarrow STOP$$

In CSP semantics, each of the possible path of events happen is named as a "trace of event". In this case, there exists four traces, including the four possible path and an empty trace that nothing happen (no event performed). One of the traces is shown below.

$$\langle Send.A.B.m1, Receive.A.B.m3, STOP \rangle$$

There exist a lot of different trace, and sometimes the traces will be in different size, there exist a kind of empty trace, which is defined as

$$\langle\rangle$$

Also, if traces(P) is a set of all possible trace for process P and A is a set of event, then tr in the following statement means the maximal trace in traces(P) set only with event in the set A.

$$tr \upharpoonright A \qquad where \;\; tr \in traces(P)$$

One of the proving and modelling direction mentioned in [Sch96] is to check if some of the trace specification is fulfil for the modelling of the communication sequence and protocol message exchange. One of the checking is done by checking if a process (and all its traces) fulfil some of the preceding rules. For example, if process P satisfy the requirement that all events in event set R precedes all events in event set T (that is if some events in T exist as part of a trace tr of process P, then there exists some events from set R also appear

in the same trace and all of them will appear before any events from set T). This trace specification P satisfy R precedes T can be illustrated by the following statement.

$$P \textbf{ sat } R \ \mathit{precedes} \ T \iff \forall tr \in \mathit{traces}(P) \bullet (tr \upharpoonright T \neq \langle\rangle \implies tr \upharpoonright R \neq \langle\rangle)$$

In some of the cases, the trace specification can use to check if some signalling message claiming the fulfilment of some security properties can exist in the trace if some violation of the security properties occurs. For example, if a trace specification mentioned that the authenticated signalling can only exist in the trace if accurate proving of the identity by message exchange happens before. If a process cannot satisfy this trace specification, then this process have potential vulnerabilities exist for an adversary to break the authentication.

5.2 Adversaries Knowledge Consideration

In the basic CSP modelling and theorem proving mentioned in [Sch96, Sch98], it only considers the communication process and all the messages exist in the communication network environment. In a general case, all the encryption and cryptographic algorithm are ignored in the modelling, and they are all treated as a black box with no flaw consideration in that part. In reality, some flaws in the protocol is coming from a bad implementation or design of the cryptographic algorithm used in the communication. Also, there exists some cases that some of the message exchange in the communication process will leak out information for the adversaries to deduce more information they need to break the protocol and security properties. In this case, the protocol itself does not leak out information directly, but it leak out part of the information that the adversaries need for gaining knowledge of some cryptographic secret value and make an attack towards the protocol itself. So the knowledge deduction and generation abilities of the adversaries also required to be analysed, because it will directly affect the adversary's knowledge of the communication network environment and thus affecting the security properties proving of the protocol.

5.2.1 Knowledge Deduction and Generation

From the original set of CSP modelling approach, there are already some consideration of the adversary knowledge like

$$\begin{aligned} &\mathit{Adversary}(K) = \\ &\forall_{i,j \in U,m} \bullet (\mathit{Send.i.j.m} \rightarrow \mathit{Adversary}(K \cup m) \ \Box \ \mathit{Receive.i.j.m} \rightarrow \mathit{Adversary}(K \cup m)) \end{aligned}$$

In the above statement, it shows that any users in the user set U sent or receive a message in the communication channel. The adversary can make use of the message send to increase its knowledge set K. In this consideration, we have no focus on what message m can help the adversaries to deduce and create more knowledge from it. Also, in the whole modelling technique, we have not kept track and model on what is required for an adversary to generate some secret value, for example, keystream or shared key. These changes will also be a part that are needed to examine in the complete view of the network environment.

In the CSP language, there exists an operator to show the generate relationship of message, but it is not enough to consider the requirements for an adversary to deduce a cryptographic secret value. The deduction rules should be included together to the network

environment because it can help to infer any possible leak of partial secret information that will lead to a flaw in the cryptographic algorithm. The following is some rules of the generation operator used in basic CSP where m is a random message and K is the adversary knowledge set and $\{\mathrm{m}\}_{\mathrm{k}}$ represent a ciphertext formed by encrypting message m by key k using symmetric encryption.

$$m \in K \implies K \vdash m$$

$$(K \vdash m \wedge K \vdash k) \implies K \vdash \{m\}_k$$

$$(K \vdash \{m\}_k \wedge K \vdash k) \implies K \vdash m$$

It is obvious that, the basic CSP approach just consider the ability of an opponent to reuse messages or the encryption and decryption of messages which all the necessary information are already in the knowledge set. It does not concern about how the messages in the knowledge set can help the adversary to generate more messages that are useful. For example, if we consider the first keystream ks_1 used in the MiFare Classic authentication protocol, then it can be modelling as follow where k means the sector key and $\mathrm{N_T}$ represent the tag nonce.

$$\text{if}(k \in K \wedge N_T \in K \wedge UID \in K) \text{ then } K \vdash ks_1$$

In this case, the requirement for an adversary to generate the secret keystream ks_1 has been modelled. And so if the adversary knows the shared key k before, then after he see the plaintext of UID and $\mathrm{N_T}$ in the first two message and add that to his knowledge set, he can also get the value of ks_1. Thus by the basic generation rule, he is also able to get the value $\mathrm{N_R}$ which is encrypted by ks_1. This method introduces the modelling and analysis of the indirect message generation of the adversary with its knowledge. The adversary statement should be changed a little bit to the following with set C include all the communication channel (send, receive and etc.) and set N which contains all message that can be generated by some knowledge of different value. For example, set N may include the generation rules for $\{\mathrm{ks}_1\ \mathrm{ks}_2\ \mathrm{ks}_3\ \mathrm{N_R}\}$

$$Adversary(K) =$$

$$\forall_{i,j \in U, n \in N, m} \bullet ((\Box_{c \in C}\, c.i.j.m \rightarrow Adversary(K \cup m) \rightarrow Adversary(K \cup (\text{if } K \vdash n \text{ then } n \text{ else } \emptyset)))$$

5.2.2 Affect to Rank Function Theorem Proving

With the new consideration of the indirect message generation and deduction, there exists great changes in the theorem proving steps. The main reason for the differences is, now we also need to keep track of the content and change of the adversary knowledge set because some indirect message generation will increase the size of the knowledge set. This setting will possibly increase the ability of an adversary to get more information or to fabricate more messages. By the rules of the rank function, message that can be seen or created by an adversary using his knowledge set should be classified as positive rank. Also, protocol and message exchange should maintain the positive rank throughout the communication. So, the indirect message generation will need to considerate because it may affect the rank function. This setting become an additional part to prove in the rank function theorem proving, and it can surely help to increase the angle and view on the whole adversary modelling and protocol analysis.

5.3 Proposed Solution

In this section, the new-proposed solution for modelling on smart transport ticketing system will be concluded here. The whole modelling approach will base on the basic CSP modelling direction proposed in [Sch96, Sch98] by Prof. Schneider. But there exists some additional things in the proposed solution. Firstly, there will be a new indirect generation set exist in the model; it will contains all the rules and requirements for indirect message generation and deduction. This set will be used to check if the adversary has the abilities to deduce more message after a message exchange occurs. Also, there will be a new set that keep track of the adversary knowledge during the modelling period. This set is necessary because, in the theorem proving consideration, the rank allocation of each message will affect by the knowledge of the opponents. So, the update of the knowledge set will also needed to be monitored. The next change is on the capabilities of the adversary. Usually, the adversaries will only gain the new message into their knowledge set. But it is not the same case in the new approach, after putting the new message into the knowledge set, an extra step of checking is needed to see if any more message can be deduced or generated. If such message exists, the knowledge will need to update again to record the change of the knowledge set from indirect message generation. Lastly, the rank function theorem proving need to consider changes in the knowledge set during the rank assignment and testing period. In the next chapter, the new-proposed approach will be illustrated on the three protocol mentioned in Chapter 4. After that, the new approach will be analysed.

6 Analysis the Proposed Adversary Model

The last chapter of the report aimed to illustrate the proposed adversary model from chapter 5 by applying it to the three protocols mentioned in chapter 3. After the modelling and the rank function theorem proving of the three protocols, a brief conclusion and analysis of the proposed protocol will be given at the end of this chapter.

6.1 Illustration of Proposed Model

At this part, the three protocols mentioned in chapter four will be modelled by the proposed adversary model. The modelling will be started by a network environment model. After that, a model of all generation and deduction of the secret value will be given. Then a process behaviour of all component of the network environment will be modelled separately. After that, the security properties and signalling consideration will be added to the model. At last, the rank function theorem proving will be given on the model. First of all, the network environment for the three protocols are the same. So the model statement is the same, which is shown as following.

$$NETWORK = (Card_A(N_T) \;|||\; Reader_B(N_R)) \underset{send,receive}{\|} Adversary(K)$$

where Card_A and Reader_B represent that it can be any card communicating with any reader in the network environment. Also, there exists an adversary with initial knowledge set K.

6.1.1 Original MiFare Classic Authentication Protocol

First of all, the deduction and generation rule of the secret value will be given as follow.

$$\text{if}(k \in K \wedge N_T \in K \wedge UID \in K) \text{ then } K \vdash ks_1$$

$$\text{if}(k \in K \wedge N_T \in K \wedge UID \in K \wedge N_R \in K) \text{ then } K \vdash ks_2 \wedge K \vdash ks_3$$

$$\text{if } ks_3 \in K \text{ then } K \vdash ks_2$$

$$\text{if}(ks_2 \in K \wedge N_R \oplus ks_1 \in K) \text{ then } K \vdash ks_1$$

$$\text{if}(ks_1 \in K \wedge UID \in K \wedge N_T \in K) \text{ then } K \vdash k$$

$$\text{if } N_T \in K \text{ then } K \vdash A_R \wedge K \vdash A_T$$

In this protocol, the secret value is the sector key k sharing by the reader and the card sector. Also, the three keystream used in the encryption algorithm is a secret value which need the knowledge of some value to produce it. The standard-generation requirement of the three keystream used is shown on the first two statements. The next three statements are given to illustrate the flaws in the Crypto1 algorithm. It can help the adversaries generate and deduce the sector key k if he processes some of the keystream in his knowledge set K. It is a possible attack towards the protocol and so it needed to be in the set of deduction and generation rules for secret value. The last statement is the rule from MiFare Classic reply that the two reply answer is directly deduced by the nonce from the tag (card).
After the network environment model, the next step is going deep into the process event of each actor. The adversaries behaviour is always the same for the three protocols and is shown below which has been explained in the last chapter.

$Adversary(K) =$

$$\forall_{i,j \in U, n \in N, m} \bullet ((\Box_{c \in C}\, c.i.j.m \rightarrow Adversary(K \cup m) \rightarrow Adversary(K \cup (\text{if } K \vdash n \text{ then } n \text{ else } \emptyset)))$$

Then the next actor will be the set of cards contain the MiFare Classic chip. It is shown as follow.

$$Card_A(N_T) = \Box_{b \in R}(Send.A.b.UID \rightarrow Send.A.b.N_T \rightarrow Receive.A.b.\{N_R\}_{ks_1} \rightarrow Receive.A.b.\{A_R\}_{ks_2}$$
$$\rightarrow Commit.A.b.(N_R, k) \rightarrow Send.A.b.\{A_T\}_{ks_3} \Box Receive.A.b.\{Halt\}_{ks_3} \rightarrow STOP)$$

In the statement above, the Commit.A.b.(N_R,k) is a signal by the card. It mentioned that at that point of time, the card A is confirmed that it is communicating with one of the readers, and the reader choose the nonce N_R and have the knowledge of the shared sector key k. So the reader is authenticated by the card as an honest reader. In reality, it is not matter that actual reader is authenticated, the card just need to know it is a genuine reader representing the transport company. At the second to last step of the process, it is possible that the reader would send a halt (represent by an integer) signal encrypted with ks_3 to the card if the card did not respond the other message before timeout. So there is a choice in that step. Either the card sends the reply to the reader to authenticate itself or the reader send halt message because of a timeout. In either step, this authentication will end and may start the communication or redo the authentication.
Next model will be the model of the reader. It is shown as following.

$$Reader_B(N_R) = Receive.B.a.UID \rightarrow Receive.B.a.N_T \rightarrow Send.B.a.\{N_R\}_{ks_1} \rightarrow Send.B.a.\{A_R\}_{ks_2}$$
$$\rightarrow (Receive.B.a.\{A_T\}_{ks_3} \rightarrow Commit.B.a.(N_T, k)) \Box Send.B.a.\{Halt\}_{ks_3} \rightarrow STOP)$$

In this statement above, the reader will only start the authentication when any card a send the UID to the reader in the last step of the anti-collision stage. so it does not have a choice operator because the reader will only communicate with the card starting this authentication. There is another signal Commit.B.a.(N_T,k) claiming that at that point of time, the reader B can confirm that it is really communicating with a card and the card generates the nonce N_T and shares the same key k which can match to the identity of the UID provided. So the card is authenticated to the reader. This signal act as a pair of signal with the one send by the card. This pair of signal proves the mutual authentication between the card and the readers. We can also state as follow.

$$NETWORK \textbf{ sat } Commit.A.B.(N_R, k) \ \ precedes \ \ Commit.B.A.(N_T, k)$$

which is similar to the situation in trace specification that R precedes T. So, in the rank function theorem proving, we can aim to prove if the network satisfy this requirement. In the opposite way, we can also prove by contradiction. That is if we force the network environment to parallel with the stop process on Commit.A.B.(N_R,k), then the network can never reach the other statement Commit.B.A.(N_T,k) because the first one precedes the second one. And so we aimed to prove that if this is the case, the maximum trace projection for Commit.B.A.(N_T,k) should be an empty trace. It can be illustrated as

$$NETWORK \underset{Commit.A.B.(N_R,k)}{\|} STOP \textbf{ sat } tr \upharpoonright Commit.B.A.(N_T, k) = \langle\rangle$$

After the modelling, we will start the rank function theorem proving. First, the rule of rank theorem mentioned in [Sch98] will be restated.

1. The rank function for all message in the initial adversary knowledge set should be positive.

2. The rank function for all message that can be deduced and generated from the adversary knowledge set should be positive.

3. The rank function for all message in set T (in the R precedes T relation) should be non-positive.

4. All participants (Reader and Card) should maintain positive rank in the communication.

If there is a rank function exists and fulfil all the above requirement, then we prove the fact R precedes T is correct. And subsequently, the mutual authentication of the protocol can be proved. First of all, the nonce of tag and the UID is sent in clear, so they are assigned with positive rank. The next thing to consider is the two signals, as we are proving the R precedes T trace specification, so Commit.A.B.(N_R,k) is in set R and Commit.B.A.(N_T,k) is in set T. And according to the rank function rules, Commit.B.A.(N_T,k) in set T will also have non-positive ranking. And R will have positive ranking. The only secret in the communication is the shared sector key, as it is not in the adversary knowledge set, so it is assigned with non-positive ranking, all other possible keys exist in the environment is assigned with positive ranking. Because we are now proving by contradiction and restricted the communication to stop at R (Commit.A.B.(N_R,k)), so all other messages after it should not appear in the environment and so all of them are assigned with a non-positive rank. So the overall rank function is defined as follows

$$\rho(N_T) = 1$$

$$\rho(UID) = 1$$

$$\rho(Commit.A.B.(N_R, k)) = 1$$

$$\rho(Commit.B.A.(N_T, k)) = 0$$

$$\rho(Key) \begin{cases} 0, & \text{if Key = sector key k} \\ 1, & \text{otherwise} \end{cases}$$

$$\rho(\{m\}_{ks}) \begin{cases} 0, & \text{if m = } A_T \text{ and ks = } ks_3 \\ & \text{or m = Halt and ks = } ks_3 \\ 1, & \text{otherwise} \end{cases}$$

$$\rho(m_1, m_2) = min(\rho(m_1), \rho(m_2))$$

We now need to use the four rank function rules to prove the protocol. For rule 1, the initial adversary knowledge set contains only all possible UID of the card but nothing more. And we define all UID with positive rank, so rule 1 is fulfil. For rule 2, it needs to consider all the possible generation of the message. As mentioned in the last chapter, the new-proposed approach needed to keep track of the change of the adversary knowledge. We will do it as follows. After message 1, adversary knows the UID of the card in the communication.

$$K = \{UID\}$$

After message 2, adversary knows the tag nonce.

$$K = \{UID, N_T\}$$

Also, because of the last generation rule, the adversary can calculate the tag and reader reply.

$$\text{if } N_T \in K \text{ then } K \vdash A_R \wedge K \vdash A_T$$

$$K = \{UID, N_T, A_T, A_R\}$$

After reader replies message 3 to the card, adversary know the encrypted reader nonce and the encrypted reader answer.

$$K = \{UID, N_T, A_T, A_R, N_R \oplus ks_1, A_R \oplus ks_2\}$$

Also, because the encryption is done by a simple xor and the adversary know both the plain text and cipher text of the reader response, so the adversary knows the keystream ks_2.

$$K = \{UID, N_T, A_T, A_R, N_R \oplus ks_1, A_R \oplus ks_2, ks_2\}$$

By the generation rule, if the encrypted version of reader nonce N_R and ks_2 is known, then the adversary can deduce ks_1.

$$\text{if}(ks_2 \in K \wedge N_R \oplus ks_1 \in K) \text{ then } K \vdash ks_1$$

$$K = \{UID, N_T, A_T, A_R, N_R \oplus ks_1, A_R \oplus ks_2, ks_2, ks_1\}$$

If the adversary knows ks_1, UID and tag nonce N_T, then he is able to deduce the shared sector key k

$$\text{if}(ks_1 \in K \wedge UID \in K \wedge N_T \in K) \text{ then } K \vdash k$$

$$K = \{UID, N_T, A_T, A_R, N_R \oplus ks_1, A_R \oplus ks_2, ks_2, ks_1, k\}$$

From the above statement, we know that the adversary can generate and deduce the shared sector key from the knowledge set after message 3. So after message 3, all the message in the adversary knowledge (include all message that can be deduced) should assign with a positive rank in order to keep the rank function rule 2 correct. But we can see that, in the rank function defined above, the shared secret key k is a secret value and should apply with non-positive ranking. This setting creates a contradiction and so it proves that the statement R precedes T is not necessary correct. This conclusion proves the protocol is insecure in some sense. The adversary can break the protocol or impersonating parties in the communication.

6.1.2 Improved Protocol in [HC12]

For the improved protocol, we also defined the generation rule as follows where TS is the timestamp choose by the tag that aimed to provide randomness of the cipher and to avoid plaintext ciphertext pair appear in the communication.

$$\text{if}(k \in K \wedge TS \in K \wedge UID \in K) \text{ then } K \vdash ks_1 \wedge K \vdash ks_2$$

$$\text{if}(N_T \in K \wedge N_R \in K) \text{ then } K \vdash ks_3 \wedge K \vdash ks_4$$

$$\text{if } N_T \in K \text{ then } K \vdash A_R \wedge K \vdash A_T$$

The first two steps of the cipher try to use a random value of the time stamp to generate two keystreams to encrypt the two nonce choose by the reader and the tag. This setting can be treated as confidential value exchange and in this case, there are no known generation rule that recover the key from the keystream as non-linear filter is used in the keystream generation. So by the current knowledge of the improved protocol, only the above rules exist to deduce secret value and keystream.
After the network environment model, the next step is going deep into the process event of each actor. As mentioned above, the adversary is still working under the same network environment and so the behaviour and process modelling is still the same

$$Adversary(K) =$$

$$\forall_{i,j \in U, n \in N, m} \bullet ((\Box_{c \in C}\, c.i.j.m \rightarrow Adversary(K \cup m) \rightarrow Adversary(K \cup (\text{if } K \vdash n \text{ then } n \text{ else } \emptyset)))$$

Then the next actor will be the set of cards contain the MiFare Classic chip, it is a little bit different than the original MiFare Classic authentication protocol. It is shown as follows.

$$Card_A(N_T, TS) = \Box_{b \in R}(Send.A.b.UID \rightarrow Send.A.b.TS \rightarrow Receive.A.b.\{N_R\}_{ks_1} \rightarrow Send.A.b.\{N_T\}_{ks_2} \rightarrow Receive.A.b.\{A_R\}_{ks_3} \rightarrow Commit.A.b.(N_R, k) \rightarrow Send.A.b.\{A_T\}_{ks_4} \rightarrow STOP)$$

Next model will be the model of the reader. It is shown as follows.

$$Reader_B(N_R) = Receive.B.a.UID \rightarrow Receive.B.a.TS \rightarrow Send.B.a.\{N_R\}_{ks_1} \rightarrow Receive.B.a.\{N_T\}_{ks_2} \rightarrow Send.B.a.\{A_R\}_{ks_3} \rightarrow Receive.B.a.\{A_T\}_{ks_4} \rightarrow Commit.B.a.(N_T, k) \rightarrow STOP)$$

In the above statement of the card and the reader, a pair of commit signal means that the actor know the communication partner is the correct partner which provide one of the nonces that are required to create keystream 3 and 4. And it also processes the shared sector key, and so mutual authentication can be provided by the protocol with that two signal. In this case, we are still concerning the security properties of mutual authentication and so we are still trying to prove the following

$$NETWORK \textbf{ sat } Commit.A.B.(N_R, k) \ \ precedes \ \ Commit.B.A.(N_T, k)$$

or prove it by an opposite way

$$NETWORK \underset{Commit.A.B.(N_R,k)}{\|} STOP \textbf{ sat } tr \upharpoonright Commit.B.A.(N_T, k) = \langle\rangle$$

Next step is the define of rank function for the theorem proving. First of all, the timestamp choose by the tag and the UID of the tag is sent in clear, so they are assigned with positive rank. The next thing to consider is the two signal, as we are proving the R precedes T trace specification, so Commit.A.B.(N_R,k) is in set R and Commit.B.A.(N_T,k) is in set T. And according to the rank function rules, Commit.B.A.(N_T,k) in set T will also have non-positive ranking. And R will have positive ranking. The only secret in the communication is the shared sector key, as it is not in the adversary knowledge set, so it is assigned with non-positive ranking, all other possible keys exist in the environment is assigned with positive ranking. Because we are now proving by contradiction and restricted the communication to stop at R (Commit.A.B.(N_R,k)), so all other messages after it should not appear in the environment and so all of them are assigned with a non-positive rank. So the overall rank function is defined as follows.

$$\rho(TS) = 1$$

$$\rho(UID) = 1$$

$$\rho(Commit.A.B.(N_R, k)) = 1$$

$$\rho(Commit.B.A.(N_T, k)) = 0$$

$$\rho(Key) \begin{cases} 0, & \text{if Key = sector key k} \\ 1, & \text{otherwise} \end{cases}$$

$$\rho(\{m\}_{ks}) \begin{cases} 0, & \text{if m = } A_T \text{ and ks = } ks_4 \\ 1, & \text{otherwise} \end{cases}$$

$$\rho(m_1, m_2) = min(\rho(m_1), \rho(m_2))$$

The next step is to check the four rank function rules to see if contradiction is occurred. For rule 1, the initial adversary knowledge set contains only all possible UID of the card

but nothing more. And we define all UID with positive rank, so rule 1 is fulfil. For rule 2, it needs to consider all the possible generation of message. Same as above, we will keep track the change of knowledge set one by one. After message 1, adversary knows the UID of the card in the communication.

$$K = \{UID\}$$

After message 2, adversary knows the timestamp choose by the tag.

$$K = \{UID, TS\}$$

After reader replies message 3 to the card, adversary knows the encrypted reader nonce.

$$K = \{UID, TS, N_R \oplus ks_1\}$$

After message 4, adversary knows the encrypted tag nonce. At this time, both the tag and reader get enough information to generate the next two keystream for the authentication.

$$K = \{UID, TS, N_R \oplus ks_1, N_T \oplus ks_2\}$$

After message 5, adversary knows the encrypted reader response. Although the reader response is still deducible from the tag nonce, but the tag nonce does not send in clear this time, so the adversary knows nothing on the reader response value.

$$K = \{UID, TS, N_R \oplus ks_1, N_T \oplus ks_2, A_R \oplus ks_3\}$$

This is the last message send before the signal that is all the messages we need to consider in the restricted case. We can conclude that, no non-positive rank message defined by the rank function can be generated or deduced from the adversary knowledge set. So rule 2 of the rank function fulfilled.
In the current situation, only the second signal send from the reader is existed in the set T of the R precedes T trace specification. And in our rank function, that signal is already assigned with a non-positive rank. So rule 3 of the rank function theorem is satisfied. For the last rule, we need to consider if all actors in the network environment main positive rank for all their message in the restricted case of set R (set contain only the first signal). In our environment, we only have three possible actors; they are card, reader and adversary. Among this three actor, only the card is limited by the first signal because the other actors don't generate that signal in their process behaviour. So they all considered as maintaining positive ranking. The only actor to consider is the Card, which is shown as follows with the restricted case.

$$Card_A(N_T, TS) = \Box_{b \in R}(Send.A.b.UID \rightarrow Send.A.b.TS \rightarrow Receive.A.b.\{N_R\}_{ks_1} \rightarrow$$

$$Send.A.b.\{N_T\}_{ks_2} \rightarrow Receive.A.b.\{A_R\}_{ks_3} \rightarrow$$

$$\text{if b = B then } STOP$$

$$\text{else } Commit.A.b.(N_R, k) \rightarrow Send.A.b.\{A_T\}_{ks_4} \rightarrow STOP)$$

From the restricted case, we can find that if a genuine reader B sends out the last message with an encrypted reader response encrypted with the keystream generated by the two nonce. Then on the tag side, it can consider the communication parties in this case is the genuine reader B. It is because only the legitimate reader B process the shared sector key and thus able to generate the keystream to decrypt the nonce and generate keystream 3 and 4. No other adversary without knowing the shared key can send that message and so

the running will always stopped before the first signal as b is always equals B if the previous statement is correctly received. So the actor of card also maintain positive ranking in the running. And so the rule 4 of the rank function theorem are also satisfied. From the rank function theorem proving, there exists a rank function that can model the restricted case of R precedes T and so it means that the condition is fulfilled, and the protocol is considered to be secure in this direction if the shared sector key is kept secret. There exists some cases that default key is used as the shared sector key. This setting directly break rule 1 because with the use of the shared sector key; the adversary's knowledge will also include that key and violating the non-positive ranking of the sector key k. Without consideration on the implementation issue, the protocol is considered as secure by the proposed approach.

6.1.3 Improved Protocol in [JK13]

For the other improved protocol, we still need to define the generation rule as follow where R_1 R_2 R_3 are the three random values that aimed to provide randomness to obscure the value of the keystream. The random value is generated recursively by an S-Box that keep feeding the last output as the next input. The first input to the S-Box is either the xor of the two sector keys or the sector key that is not used in the keystream 1 generation. The two shared sector keys used is illustrated as k_1 k_2.

$$\text{if}(k_1 \in K \wedge N_{T1} \in K \wedge UID \in K) \text{ then } K \vdash ks_1$$

$$\text{if}(N_{R1} \in K \wedge R_1 \in K) \text{ then } K \vdash ks_2$$

$$\text{if}(N_{R2} \in K \wedge R_2 \in K) \text{ then } K \vdash ks_3$$

$$\text{if}(k_1 \in K \wedge k_2 \in K) \text{ then } K \vdash R_1$$

$$\text{if } R_1 \in K \text{ then } K \vdash R_2$$

$$\text{if } R_2 \in K \text{ then } K \vdash R_3$$

$$\text{if } N_T \in K \text{ then } K \vdash A_R \wedge K \vdash A_T$$

In this protocol, the real value of the keystream is masked by the recursive output value of the S-Box, so it also don't have known way to recover the state of the LFSR and thus recovering the shared sector key. So the above rules just concentrating on the generation of key and random value of S-Box.
After the network environment model, the next step is going deep into the process event of each actor. Again, the adversary is still working under the same network environment and so the behaviour and process modelling is still the same

$$Adversary(K) =$$

$$\forall_{i,j \in U, n \in N, m} \bullet ((\Box_{c \in C}\, c.i.j.m \rightarrow Adversary(K \cup m) \rightarrow Adversary(K \cup (\text{if } K \vdash n \text{ then } n \text{ else } \emptyset)))$$

Then the next actor will be the set of cards contain the MiFare Classic chip, it is a little bit different and is shown as follows.

$$Card_A(N_{T1}, N_{T2}) = \Box_{b \in R}(Send.A.b.UID \rightarrow Send.A.b.N_{T1} \rightarrow Receive.A.b.\{N_{R1}\}_{ks_1 \oplus R_1} \rightarrow$$

$$Receive.A.b.\{N_{R2}\}_{ks_2 \oplus R_2} \rightarrow Commit.A.b.(N_{R1}, N_{R2}, k_1, k_2) \rightarrow Send.A.b.\{N_{T2}\}_{ks_3 \oplus R_3} \rightarrow STOP)$$

Next model will be the model of the reader. It is shown as follows.

$$Reader_B(N_{R1}, N_{R2}) = Receive.B.a.UID \rightarrow Receive.B.a.N_{T1} \rightarrow Send.B.a.\{N_{R1}\}_{ks_1 \oplus R_1} \rightarrow$$
$$Send.B.a.\{N_{R2}\}_{ks_2 \oplus R_2} \rightarrow Receive.B.a.\{N_{T2}\}_{ks_3 \oplus R_3} \rightarrow Commit.B.a.(N_{T1}, N_{T2}, k_1, k_2) \rightarrow STOP)$$

In the above statement of the card and the reader, the pair of commit signal means that the actor knows the communication partner is the correct partner which provide two nonce for creating key and authentication. And it also processes the two shared sector key which is used to generate the keystream and the S-Box output recursively. So mutual authentication can be provided by the protocol with that two signal. In this case, we are still concerning the security properties of mutual authentication and so we are still trying to prove the following.

$$NETWORK \textbf{ sat } Commit.A.B.(N_{R1}, N_{R2}, k_1, k_2) \ \ precedes \ \ Commit.B.A.(N_{T1}, N_{T2}, k_1, k_2)$$

or prove it by an opposite way.

$$NETWORK \underset{Commit.A.B.(N_{R1},N_{R2},k_1,k_2)}{\parallel} STOP \textbf{ sat } tr \upharpoonright Commit.B.A.(N_{T1}, N_{T2}, k_1, k_2) = \langle\rangle$$

Next step is the creation of the rank function. First of all, the first tag nonce and the UID of the tag is still sent in clear, so they are assigned with positive rank. The next thing to consider is the two signals, as we are proving the R precedes T trace specification, so Commit.A.B.(N_{R1},N_{R2},k_1,k_2) is in set R and Commit.B.A.(N_{T1},N_{T2},k_1,k_2) is in set T. And according to the rank function rules, Commit.B.A.N_{T1},N_{T2},k_1,k_2 in set T will also have non-positive ranking. And R will have positive ranking. The only secret in the communication is the two shared sector key k_1 and k_2, as it should not in the adversary knowledge set, so it is assigned with non-positive ranking, all other positive keys exist in the environment is assigned with positive ranking. Because we are now proving by contradiction and restricted the communication to stop at R (Commit.A.B.N_{R1},N_{R2},k_1,k_2), so all other messages after it should not appear in the environment and so all of them are assigned with a non-positive rank. So the overall rank function is defined as follows

$$\rho(N_{T1}) = 1$$

$$\rho(UID) = 1$$

$$\rho(Commit.A.B.(N_R, k)) = 1$$

$$\rho(Commit.B.A.(N_T, k)) = 0$$

$$\rho(Key) \begin{cases} 0, & \text{if Key = sector key } k_1 \\ & \text{or Key = sector key } k_2 \\ 1, & \text{otherwise} \end{cases}$$

$$\rho(\{m\}_{ks}) \begin{cases} 0, & \text{if m = } N_{T2} \text{ and ks = } ks_3 \oplus R_3 \\ 1, & \text{otherwise} \end{cases}$$

$$\rho(m_1, m_2) = min(\rho(m_1), \rho(m_2))$$

The next step is to check the four rank function rules to see if contradiction occurs. For rule 1, it is the same as the above two cases, all adversary initial knowledge has been set to positive rank and so rule 1 is satisfied. For rule 2, it needs to consider all the possible generation of the message. Same as above, we will keep track the change of knowledge set one by one. After message 1, adversary knows the UID of the card in the communication.

$$K = \{UID\}$$

After message 2, adversary knows the first tag nonce.

$$K = \{UID, N_{T1}\}$$

After reader replies message 3 to the card, adversary knows the two encrypted reader nonce.

$$K = \{UID, TS, N_{R1} \oplus ks_1 \oplus R_1, N_{R2} \oplus ks_2 \oplus R_2\}$$

After message 4, adversary knows the second tag nonce in encrypted form.

$$K = \{UID, TS, N_{R1} \oplus ks_1 \oplus R_1, N_{R2} \oplus ks_2 \oplus R_2, N_{T2} \oplus ks_3 \oplus R_3\}$$

This statement is the last message send before the signal that is all the message we needed to consider in the restricted case. We can conclude that, no non-positive rank message defined by the rank function can be generated or deduced from the adversary knowledge set. So rule 2 of the rank function is fulfilled.
In the current case, only the second signal send from the reader is existed in the set T of the R precedes T trace specification. And in our rank function, that signal is already assigned with a non-positive rank. So rule 3 of the rank function theorem is satisfied. For the last rule, same as the last protocol, only the card is restricted by the first signal because the other actors don't generate that signal in their process behaviour. So they all considered as maintaining positive ranking. The only actor to consider is the Card, which is shown as follows with the restricted case.

$Card_A(N_{T1}, N_{T2}) =$

$\Box_{b \in R}(Send.A.b.UID \rightarrow Send.A.b.N_{T1} \rightarrow Receive.A.b.\{N_{R1}\}_{ks_1 \oplus R_1} \rightarrow Receive.A.b.\{N_{R2}\}_{ks_2 \oplus R_2} \rightarrow$

if b = B then $STOP$

else $Commit.A.b.(N_{R1}, N_{R2}, k_1, k_2) \rightarrow Send.A.b.\{N_{T2}\}_{ks_3 \oplus R_3} \rightarrow STOP)$

From the restricted case, we can find that if a genuine reader B sends out the last message with two encrypted nonce which the keystream and the S-Box value is generated recursively from the two shared sector keys separately. Then the tag can sure that the communication party in this case can only be a legitimate reader B who also shared the same secret value (two sector keys). No other adversary without knowing the shared keys can send that message because the generation of any keystreams and random values need to be done recursively from the shared key value. As a result, the running will always stopped before the first signal in the limited case as b is always equals B if the previous statement is correctly received. So the actor of card also maintain positive ranking in the running. For a conclusion, rule 4 of the rank function theorem are also satisfied. From the rank function theorem proving, there exists a rank function that can model the restricted case of R precedes T and so it means that the condition is fulfilled, and the protocol is considered to be secure in this direction if the shared sector keys are kept secret. Without consideration on the implementation issue, the protocol is considered as secure by the proposed approach.

6.2 Analysis of the Proposed Model

From the illustration and practical modelling of the proposed approach on the three protocols. We can conclude that, possible attacks can be found for the original MiFare Classic authentication protocol. Normally, in the formal adversaries modelling and protocol analysis, the flaw in the cryptographic algorithm will be ignored. By this new approach, we can add the consideration of some known attack on a cryptographic algorithm into the modelling and protocol analysis in order to make the formal modelling closer to the real situation. It maybe not practical to mentioned some cryptographic algorithm which the

details are not known. But in the real world, many of the protocol design start to consider the use of some public cryptographic algorithm. In this case, putting the flaws and consideration of the generation and deduction of message from cryptographic algorithm to the formal modelling is also a possible and practical way to get a whole picture of the security level of the protocol. This approach can put additional consideration on the whole network environment in additional to the communication and message exchange itself. It can help to analyse what can the adversary creates or deduces from their knowledge set in addition to the information and data directly leak out from the communication channel. This approach is a more mature way to consider the network environment. This approach also has its limitation; it can only consider some known attacks and flaws in the known cryptographic algorithm. As a result, if all the algorithm details are obscured or kept as business secret. This approach work no different than the basic CSP modelling approach mentioned by Prof. Schneider in [Sch96, Sch98].

7 Conclusion

In this project, all the objectives stated in the beginning has been achieved. The primary target of this project is to find a new approach for formal modelling of the smart transport ticketing system, and it is successfully achieved. In particular, each of the following objectives are restated below.

1. Find and propose a symbolic adversary modelling framework that are suitable for the analysis of the smart transport ticketing system and protocol.

   We had analysis two different kinds of symbolic adversary modelling approach and proposed the process algebra language of CSP as the base of the symbolic adversary modelling approach to be used in the formal modelling of the smart transport ticketing system.

2. Investigate the ways to extend the model to allow it become possible to analysis some non-logical part of the smart transport ticketing system and protocol.

   We extend the model to consider on the flaws on the cryptographic algorithm (logical part) and the change of the adversary knowledge set (non-logical part). So the new approach is extending on the basic CSP approach, with a more thorough consideration on the whole network communication environment.

3. Use formalized model to analyse smart transport ticketing system to help to identify the potential problems, weaknesses and vulnerabilities of the system and protocol, which can help to provide a solution to mitigate attacks.

   The new approach has been used to illustrate on the original MiFare Classic authentication protocol and the two improved protocols proposed by different researchers. Also, the security level of those protocols has been analysed by the rank function theorem proving approach.

4. Analysis the feasibility of the proposed model on smart transport ticketing system.

   The result of the illustration has been briefly analysed and it provides more thorough result of the security of the target protocol that are aimed to use on smart transport ticketing system.

From the study of formal symbolic adversary modelling and the MiFare Classic protocols, it is clear that with the new approach, more problems on the network running environment can be considered in additional to the message exchange on the communication. This approach can discover more problems and so it appears that the proposed approach can help in the formal modelling of the smart transport ticketing system to analyse the security properties of it.

We proposed a direction for future work. With the newly proposed adversary modelling approach, we have considered the change of the knowledge set of an adversary which is based on the known attacks and flaws of the cryptographic algorithm. It can be further investigate on the possibility of estimating the possible knowledge change and adversaries behaviour on some unknown or seemingly secure cryptographic algorithm. Also, further investigation can be done on the possibility of including the effect of environment variable and side channel attacks on the adversary behaviour and change of their knowledge set.

References

[BAN90] Michael Burrows, Martin Abadi, and Roger Needham. Logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.

[BC10] David Basin and Cas Cremers. Modeling and analyzing security in the presence of compromising adversaries. In *Computer Security–ESORICS 2010*, pages 340–356. Springer, 2010.

[Bla12] B. Blanchet. *Security protocol verification: Symbolic and computational models*, volume 7215 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2012.

[Cou09] Nicolas T Courtois. The dark side of security by obscurity and cloning mifare classic rail and building passes, anywhere, anytime. 2009.

[Cri09] Toma Cristian. Secure mechanisms for e-ticketing system. *Journal of Mobile, Embedded and Distributed Systems*, 1(2):81–94, 2009.

[dKGHG08] Gerhard de Koning Gans, Jaap-Henk Hoepman, and Flavio D Garcia. A practical attack on the mifare classic. In *Smart Card Research and Advanced Applications*, pages 267–282. Springer, 2008.

[DM99] Nancy A Durgin and DC Mitchell. Analysis of security protocols. *NATO ASI SERIES F COMPUTER AND SYSTEMS SCIENCES*, 173:369–394, 1999.

[DY83] Danny Dolev and Andrew C Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198–208, 1983.

[For03] Formal Systems (Europe) Ltd. *FDR2 User Manual*, 2003.

[GdKGM+08] Flavio D Garcia, Gerhard de Koning Gans, Ruben Muijrers, Peter Van Rossum, Roel Verdult, Ronny Wichers Schreur, and Bart Jacobs. Dismantling mifare classic. In *Computer Security-ESORICS 2008*, pages 97–114. Springer, 2008.

[GvRVS09] Flavio D Garcia, Peter van Rossum, Roel Verdult, and Ronny Wichers Schreur. Wirelessly pickpocketing a mifare classic card. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 3–15. IEEE, 2009.

[HC12] Kuo-Tsang Huang and Jung-Hui Chiu. Secured rfid mutual authentication scheme for mifare systems. *International Journal of Network Security & Its Applications*, 4(6), 2012.

[Hea00] J.A. Heather. *"Oh! Is it really you?" - Using rank functions to verify authentication protocols*. PhD thesis, Royal Holloway, University of London, 2000.

[Hoa85] C. A. R. Hoare. *Communicating Sequential Processes (Prentice-Hall International Series in Computer Science)*. Prentice Hall, 1985.

[HP03] Joseph Y Halpern and Riccardo Pucella. Modeling adversaries in a logic for security protocol analysis. In *Formal Aspects of Security*, pages 115–132. Springer, 2003.

[ISO09a] ISO15408. Information technology – security techniques – evaluation criteria for it security, 2009.

[ISO09b] ISO15693. Identification cards – contactless integrated circuit cards – vicinity cards, 2009.

[ISO11] ISO14443. Identification cards - contactless integrated circuit(s) cards - proximity cards, 2011.

[JK13] Moonseog Jun Jungho Kang, Hyungjoo Kim. Study on security scheme for mifare classic. *International Journal of Advancements in Computing Technology*, 5(13), 2013.

[Low96] Gavin Lowe. Breaking and fixing the needham-schroeder public-key protocol using fdr. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 147–166. Springer, 1996.

[Low98] Gavin Lowe. Casper: A compiler for the analysis of security protocols. *Journal of Computer Security*, 6(1-2):53–84, 1998.

[MC10] Keith E Mayes and Carlos Cid. The mifare classic story. *Information Security Technical Report*, 15(1):8–12, 2010.

[Mea91] Catherine Meadows. A system for the specification and analysis of key management protocols. In *Proceedings of the Symposium on Security and Privacy*, pages 182–195, 1991.

[NXP11a] NXP. *MIFARE Classic 1K - Mainstream Contactless smart card IC Specification*, 2011.

[NXP11b] NXP. *MIFARE Classic 4K - Mainstream Contactless smart card IC Specification*, 2011.

[Pro99] The Common Criteria Project. Commuon criteria for information technology security evaluation, August 1999.

[Ros97] A. Roscoe. *Theory and Practice of Concurrency*. Prentice Hall, 1997.

[RSG+00] Peter Ryan, Steve Schneider, Michael Goldsmith, Gavin Lowe, and Bill Roscoe. *Modelling & Analysis of Security Protocols*. Addison-Wesley Professional, 2000.

[Sch96] Steve Schneider. Security properties and csp. In *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, pages 174–187. IEEE, 1996.

[Sch98] Steve Schneider. Verifying authentication protocols in csp. *Software Engineering, IEEE Transactions on*, 24(9):741–758, 1998.

[Sch99] Steve Schneider. *Concurrent and Real-time Systems: The CSP Approach*. Wiley, 1999.

[SD05] S. Schneider and R. Delicata. Verifying security protocols: An application of csp. In *Lecture Notes in Computer Science*, volume 3525, pages 243–263, 2005.

[Sel03] Peter Selinger. Models for an adversary-centric protocol logic. *Electronic Notes in Theoretical Computer Science*, 55(1):69–84, 2003.

[Son12] Sony. *FeliCa IC Card User Manual*, 2.0 edition, 2012.

[SVRG+08] Ronny Wichers Schreur, Peter Van Rossum, Flavio Garcia, Wouter Teepe, Bart Jacobs, Gerhard De Koning Gans, Roel Verdult, Ruben Muijrers, Ravindra Kali, and Vinesh Kali. Security flaw in mifare classic. 2008.

[Tan09] Wee Hon Tan. Practical attacks on the mifare classic. *Imperial College London*, 2009.